1994

Tracking Dynamic Features in Image Sequences.

Sankar Krishnamurthy
Louisiana State University and Agricultural & Mechanical College

TRACKING DYNAMIC FEATURES IN IMAGE SEQUENCES

A Dissertation

Submitted to the Graduate Faculty of
The Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Computer Science

by
Sankar Krishnamurthy
B.S., University of Ranchi, 1986
M.S. in Engineering Science, Louisiana State University, 1991
M.S. in Systems Science, Louisiana State University, 1994
December 1994

To my Uncle and Father

Acknowledgements

I sincerely thank Prof. S. Sitharama Iyengar for the continuous support and encouragement given to me during my stay at LSU. I enjoyed working with Dr. Ronald Holyer and Matthew Lybanon.

Dr. Krishnakumar, Dr. Sridhar, Dr. Sivakumar, Dr. Hegde and Dr. Rajanarayanan helped me in various forms. I owe a lot to Raghu (Ennu Bekka) and Ramana (Kaavala) for being on my side during the gruesome PhD years.

I always remember Venky for making me comfortable at USL and the Srivastava family at the Hooper road. My special thanks to Vibs for drilling some basic concepts into my head. I cherish the good times I had with Dr. Daryl, Dr. Graham, Ranga, Sita, Sunitha, Leela, Ganesh, Loga, Dipti and Paul.

My mom faced too many downs and very little ups in her life. The only things that keep her going are patience and optimism. May god bless her with good health at all times.

I sincerely apologize to friends left out, but I always remember them. I wish very good luck to everyone.

Sankar Krishnamurthy

Contents

Dedication ii

Acknowledgments iii

Abstract vi

1 Introduction 1

1.1 Problem Domain 2
1.2 Motivation 5
1.3 Research Contributions 6
1.4 Organization of the Dissertation 7
1.5 Digital Image Processing - Introduction 7
1.6 Edge Detection in Oceanographic Images 15

2 Morphological Edge Detectors 21

2.1 Preliminary Concepts 22
2.2 Blur-Minimization Morphological Operator 24
2.3 Alpha-Trimmed Morphological Operator 25
2.4 Motivation and Scope 26

3 Histogram-Based Morphological Edge Detector 28

3.1 HMED Algorithm 33
3.2 Handling of Cloud Cover 35
3.3 Implementation Results 36
3.4 Parallel HMED Implementation 49
3.4.1 Motivation for Parallel Processing 51
3.5 Distributed HMED 54

4 Feature Labeling Techniques **58**
4.1 Introduction 58
4.2 Neural Networks 62
4.3 Relaxation Technique 63
4.4 Expert System 70

5 **76**
5.1 Topographic Classification Scheme 80
5.2 Labeling the Gulf Stream 81
5.3 Labeling of Warm Eddies 84
5.4 Implementation Results 86

6 Conclusions and Future Directions **94**

Bibliography **99**

Vita **103**

Abstract

This dissertation deals with detecting and tracking dynamic features in image sequences using digital image analysis algorithms. The tracking problem is complicated in oceanographic images due to the dynamic nature of the features. Specifically, the features of interest move, change size and shape.

In the first part of the dissertation, the design and development of a new segmentation algorithm, Histogram-based Morphological Edge Detector (HMED), is presented. Mathematical morphology has been used in the past to develop efficient and robust edge detectors. But these morphological edge detectors do not extract weak gradient edge pixels, and they introduce spurious edge pixels. The primary reason for this is due to the fact that the morphological operations are defined in the domain of a pixel's neighborhood. HMED defines new operations, namely H-dilation and H-erosion, which are defined in the domain of the histogram of the pixel's neighborhood. The motivation for incorporating the histogram into the dilation and erosion is primarily due to the rich information content in the histogram compared to the one available in the pixel's neighborhood. As a result, HMED extracts weak gradient pixels while suppressing the spurious edge pixels. An extensive comparision of all

morphological edge detectors in the context of oceanographic digital images is also presented.

In the second part of the dissertation, a new augmented region and edge segmentation technique for the interpretation of oceanographic features present in the AVHRR image is presented. The augmented technique uses a topography-based method that extracts topolographical labels such as concave, convex and flat pixels from the image. In this technique, first a bicubic polynomial is fitted to a pixel and its neighborhood, and topolographical label is assigned based on the first and second directional derivatives of the polynomial surface. Second, these labeled pixels are grouped and assembled into edges and regions. The augmented technique blends the edge and region information on a proximity based criterion to detect the features. A number of experimental results are also provided to show the significant improvement in tracking the features using the augmented technique over other previously designed techniques.

Chapter 1

Introduction

Remote sensing systems are some of the most prolific sources of digital data in the field of image analysis and understanding. Remote sensing and aerial imagery have wide applications in geological and soil mapping, land use, land cover, agriculture, oceanography, water resources planning and in other areas. In most of these applications, sensors mounted either on satellites or in low flying aircraft provide the digital data (usually imagery) of the scene under study. The problem domain to which these digital images are applied has correspondingly increased in scope and magnitude.

Image analysis techniques have been used extensively for automated interpretation of digital imagery. However, current image analysis techniques rely on human interpretation of the satellite imagery. Human interpretation is obviously varied in its level of expertise and is highly labor-intensive. With the proliferation of digital and satellite data and the increasing cost of manual interpretation, its becomes highly desirable, for certain applications, to move toward a capability for automated inter-

pretation. Digital analysis provides capabilities for making faster and much more sophisticated interpretation than is possible with the manual approach. Researchers have shown that it is both feasible and computationally practicable to develop automated vision systems to perform on a machine that task which the human vision system appears to perform effortlessly [12, 31, 30]. However, most of the vision system outperformed the human vision systems in numerical computations, but not in the interpretation of the images for reasons due to lack of efficient image processing algorithms - that works on all image settings - and knowledge representation tools. Thus there is a continuing need for efficient machine implemented analysis of IR data using digital image processing and artificial intelligence techniques. We discuss issues related to the development of an vision system to segment and interpret oceanographic features present in Infrared (digital) images in succeeding sections. The feratures in the oceanographic image change size, position and shape with time as explained below.

1.1 Problem Domain

Figure 1 is an Infrared (IR) image of the ocean obtained from the Advanced Very High Resolution Radiometer (AVHRR) abroad the NOAA-7 satellite. Such images are widely used for the study of ocean dynamics. In this image dark shades represent warmer temperatures and light shades represent colder temperatures. This image is unusually free of clouds and noise caused by atmospheric humidity. The Gulf Stream, cold eddies, and warm eddies (normally circulating at the north of the Gulf Stream)

are examples of "mesoscale" ocean features with dimensions of the order of 50-300 km.

The Gulf Stream is warmer than the Surgasso sea to its south, and much warmer than the waters to its north. Thus, its boundaries are easily detectable as edges produced by the temperature gradients in satellite IR images [54]. Sometimes clouds obscure oceanographic features, making their detection difficult. The movement of these features compounds the problems associated with the detection. For instance, the Gulf Stream drifts drastically while meandering. Sometimes these meanders lead to the "birth" of a the Gulf Stream ring, which is a special type of eddy that forms from a cut-off Gulf Stream meander [7, 43, 50]. When the Gulf Stream closes on itself, surrounding a mass of cold water at its southern boundary, a counterclockwise-rotating cold ring forms. Similarly, when the Gulf Stream surrounds a mass of warm water at its northern boundary, a clockwise-rotating warm ring originates.

Cold eddies are typically 150-300 km in diameter, while warm rings have diameters of around 100 km [27]. A warm eddy moves generally southwest at 3-8 cm/sec, with a mean lifetime of about half a year, and "dies" by coalescing with the Gulf Stream [35, 43]. A cold ring drifts southwestward at 5-10 cm/sec while not interacting with the Gulf Stream, with a mean lifetime of 1.2-1.5 years, and ultimately coalesces with the Gulf Stream [43, 46]. Since satellite IR images of the ocean often depict the mesoscale features clearly, Advanced Very High Resolution Radiameter (AVHRR) imagery is used extensively to study them. The study of the oceanographic features provides useful information on ocean dynamics and oil spills, and saves millions of dollars by facilitating navigation through the features. However, automated detection

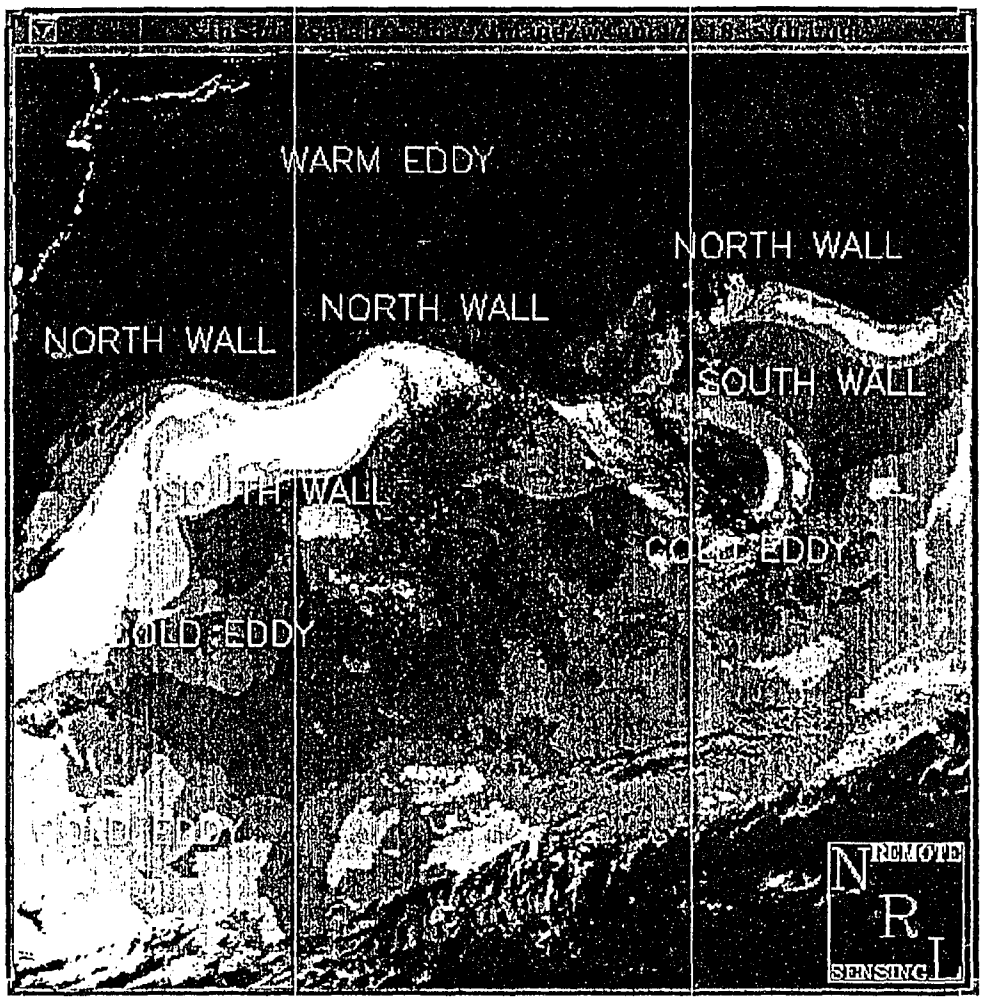


Figure 1: North Atlantic image obtained on April 17

of these features have attracted the attention of researchers in the field of computer vision and digial image analysis primarily due to lack of reliable automated image analysis techniques to extract weak gradients pertaining to cold eddies and edge and region combining algorithms to recognize the features.

1.2 Motivation

The objective of this dissertation is to develop a powerful automatic image interpretation system for oceanographic satellite images. More precisely, given an infrared image, the system should identify the features with minimal interaction from the user. The previous edge detection techniques are quite complicated in design and computationally intensive. Morphological techniques developed in the past, are simple in design and easy to construct. Moreover, morphological techniques have not been used and tested for oceanographic images. Thus, we primarily focus on the design and construction of morphological techniques to extract the weak gradients. The previous labeling (or recognition) techniques provide results which are very biased to the previous analysis primarily because the techniques operate on the edges only. We stress that new medhods need to be developed to combine edge and region information in an effective way to reduce the bias.

1.3 Research Contributions

The first part of the dissertation envisages the shortcomings of the traditional morphological operators (explained in chapter 2) in the context of edge detection. We designed and implemented a new algorithm **Histogram-Based Morphological Edge Detector** that overcomes the shortcomings. The new algorithm bridges two edge detection theories: edge detection by histogram and edge detection by mathematical gray scale morphology. We also show that the traditional morphological edge detectors are only a class of histogram-based morphological edge detectors. Our technique is applied to oceanographic satellite data and the results are compared with two other well known algorithms.

In the second part of the dissertation, a **Topography-Based Feature Labeling** technique is presented. In this method, a general computational framework is designed to address the problem of labeling oceanographic images. The essential ideas stem from fitting a bicubic polynomial to each pixel's neighborhood and assigning topological labels based on the first and second directional derivatives of the polynomial surface. The relationship between the oceanographic features in infrared satellite imagery and the topographic structures is also designed. Algorithms are developed that demonstrate ability to locate and identify the North and South Walls of the Gulf Stream and to find approximate centers of Warm and Cold eddies. Experimental results on detecting these oceanographic features are also provided.

1.4 Organization of the Dissertation

A concise overview of digital image processing and the edge detection techniques in the context of oceanographic images is given in succeeding subsections. In chapter 2 we bring the preliminary concepts and previous morphological techniques. In chapter 3 we design and implement the histogram-based morphological techniques, including the parallel and distributed implementation. The feature labeling techniques in the context of oceanographic data is discussed in chapter 4. In chapter 5 the Topography-based feature labeling technique is designed and implemented. We conclude with the future goals in chapter 6.

1.5 Digital Image Processing - Introduction

Digital image processing is basically concerned with computer processing of pictures (or images) that have been converted into a numeric form. It is a process of extracting, characterizing and interpreting information from images of a three-dimensional world. This process is commonly divided into six principal areas: (1) Sensing, (2) Preprocessing (3) Segmentation (4) Description (5) Recognition (6) Interpretation. Sensing is a process that yields a visual image. Preprocessing deals with techniques such as noise reduction and enhancement of details. Segmentation is a process that partions an image into objects of interest. Description deals with the computation of features suitable for differentiating one type of object from another. Recognition is the process that identifies these objects (e.g., wrench, bolt, nut, wheel). Finally, interpretation assigns meaning to an ensemble of recognized objects. Sometimes,

these subdivisions are grouped under three categories: Low-level, Medium-level and High-level vision. For instance, sensing and preprocessing are grouped under low-level vision, segmentation, description and recognition are grouped under medium-level vision, and interpretation is grouped under high-level vision. Most of the vision systems are implemented based on these three levels of vision. In general, the purpose of digital image processing is to enhance or improve the image in some way, or to extract information from it.

An image is divided into small regions called picture elements, or *pixels* for short. The most common subdivision scheme is the rectangular sampling grid. The image is divided into horizontal lines made up of adjacent pixels. At each pixel location, the image brightness is sampled and quantized. This step generates an integer at each pixel representing the brightness or darkness of the image at that point. When this has been done for all pixels, the image is represented by a rectangular array of integers. Each pixel has a location or address (line or row number and sample or column number) and an integer value called the gray level. This array of digital data is now a candidate for computer processing.

Digital image processing starts with one image and produces a modified version of that image. It is a process that takes an image into an image. Digital Image analysis is taken to mean a process that takes a digital image into something other than a digital image, such as a set of measurement data.

Digitizing is the process of converting an image from its original form into digital form. The term conversion is used in a nondestructive sense because the original image is not destroyed but is used to guide the generation of a digital image.

Scanning is the selective addressing of specific locations within the domain of an image. Each of the small sub-regions addressed in the scanning process is called a picture element, or pixel for short. The term scanning is loosely taken as an equivalent to the term digitizing. The rectangular grid scanning pattern is known as a raster.

The notion of contrast refers to the amplitude of gray level variations within an image. Noise is broadly defined as an additive (or multiplicative) contamination of an image. The sampling density of a digital image is the number of sample points per unit measure in the domain. Gray Scale resolution is the number of gray levels per unit measure of image amplitude. Magnification refers to the size of relationship between an image and object or image it represents.

The operations that can be performed on digital images fall into several classes. An operation is a global operation if it is applied equally throughout the entire digital image. A point operation is one in which the output pixel value depends only on the value of the corresponding input pixel. Point operations are sometimes called contrast manipulation or stretching. A local operation is one in which the output pixel value depends on the pixel values in a neighborhood of the corresponding input point.

In an image of a scene there are a variety of sources of low-level information available which can be used to form an initial description of the structure of the image. The extraction of this description, in terms of significant image tokens, is an important precursor to the construction of a more abstract description at the semantic level. It is unlikely that any single descriptive process will produce a description that is adequate for an unambiguous interpretation of the image.

Many segmentation algorithms or low-level processes use some form of smoothing operation as a preprocessing stage. Image data are complicated by errors of approximation due to the discrete nature of the representation, noise intrinsic to the sensors, and variations in the scene itself. Most simple smoothing operators do not remove noise without destroying some of the fine structure in the underlying image.

One of the simplest and most useful tools in digital image processing is the gray level **histogram**. This function summarizes the gray content of an image. The gray level histogram is a function showing the number of pixels in the image that have a particular gray value. The global histogram for many images tend to be gaussian in shape, and the area under the curve is the total number of pixels in the image. It is noted here, when an image is condensed into a histogram, all spatial information is lost. The histogram specifies the number of pixels having a particular gray level but gives no hint as to where those pixels are located within the image. Thus the histogram is unique for any particular image, but the reverse is not true.

The global histogram may not be very useful for extracting vital cues from the image because the global information will not accurately reflect local image events that do not involve large numbers of pixels. Much of the focus of the algorithms will be to organize the segmentation process around local histograms from local windows and then have a postprocessing stage merge regions that have been arbitrarily split along the artificial window boundaries. Histogram is useful for enhancing the image, changing the contrast of the image, thresholding the image.

Thresholding is process of segmenting objects of interest from the background of the image based on the gray value of the pixels in the image. Suppose we have

a light object on a dark background. One obvious way to segment the object is to choose a threshold (or gray) value T, such that a pixel whose gray value is less than T can be background pixel and if it greater than T, it is a pixel belonging to the object. Histogram is useful for thresholding operation. Multilevel thresholding is also useful if the image has one or more different objects in the image. In other words, we choose different values T1, T2, etc, such that pixel value greater than T2 belong to one object , pixel value less than T1 belong to a background, pixel value greater than T1 and less than T2 may belong to another object.

A classical problem in image processing is the detection of sudden changes in gray values from one pixel to another adjacent pixel. Such changes usually indicate a boundary, i.e., an edge, between two distinctly different objects in the image. If the difference in gray values between two adjacent pixels vary considerably, then the pixels are named as edge pixels. This method of detecting edges is called **edge detection** by thresholding. The detection of these edges is very important for analyizing the objects in the image. Moreover, this operation reduces the amount of data present in the image. In other words, previously, we have huge amount of pixels in the original image. After this operation, we have only pixels that belong to the objects of interest.

Most of the automated vision systems that have been developed for the image interpretation partitions the entire process into manageable pipelined subtasks with or without feedback. As mentioned earlier, a traditional conceptual classification divides an image analysis task into three basic modules: low, medium and high level analysis. The purpose of this type of classification of the vision system is to decompose the overall complexity of the problem into manageable portions. However, there is

no fixed dividing lines in the sequence of operations from image sensing through preprocessing, boundary detection, region growing and interpretation.

The low level analysis tasks preprocess and segment the image, and extract statistical values, texture, tone from the image without using any domain-specific knowledge. The process of partitioning an image into primitives is defined as segmentation [53]. The choice of one segmentation algorithm over another is dictated mostly by the characteristics of the problem being considered. Segmentation has become a prerequite for the description of an image. The main idea behind segmentation step is not only to reduce the massive amount of pixel data but also to extract local parameters to interpret the image.

For example, each pixel has three parameters associated with it: x and y coordinates, and an intensity value typically in the range 0-255. A pixel does not have any other information about the objects in the image. It is extremely difficult to interpret a 512 x 512 image, using only the information provided by the pixels; they provide a great deal of information, but not in a useful form. It is necessary to reduce the quantity of data without losing any structural information contained in the image. In transforming a raw image to a more reduced data format, a number of parameters are extracted namely, statistical data, texture, tonal values and many others.

This process of extracting parameters is widely known as low-level image analysis. In the last two decades, a number of low-level image processing algorithms for segmenting a raw image have been developed. However, none of the image processing techniques work in all image settings. Some techniques are sensitive – to noise in the image, to the light condition under which the objects have been sensed, to

threshold parameters and to other parameters. The common problem with the techniques is that some important features are not extracted and/or erroneous features are detected.

Sometimes image processing techniques require some kind of domain-specific knowledge for them to provide reliable results. Techniques of this type make up the intermediate level. Region growing, shape analysis, probabilistic relaxation, multispectral classification are examples of medium-level segmentation algorithms. The amount of knowledge incorporated in these techniques depends on the objects in the scene and effectiveness of the techniques.

The high level tasks unambiguously apply meaningful labels to objects in the images using shape, size, position and other information provided by the medium level modules. All domain-specific knowledge available is used to label the images. For instance, tone, texture, interrelationship between the features, and the seasonal variations and the dynamic nature of the feature, if any, are used in the high level module. A number of high level vision techniques have been reported in recent years to describe and interpret the output of the lower-level modules. The representation, use, and the acquisition of the knowledge constitutes the heart of any automated vision systems. With the fusion of Artificial Intelligence techniques in the computer vision, knowledge representation led to a shift from the procedural to declarative forms. Production rules, semantic nets, frames, blackboard architectures are some of the well known and widely used forms of knowledge representation and inference mechanisms of high-level vision techniques.

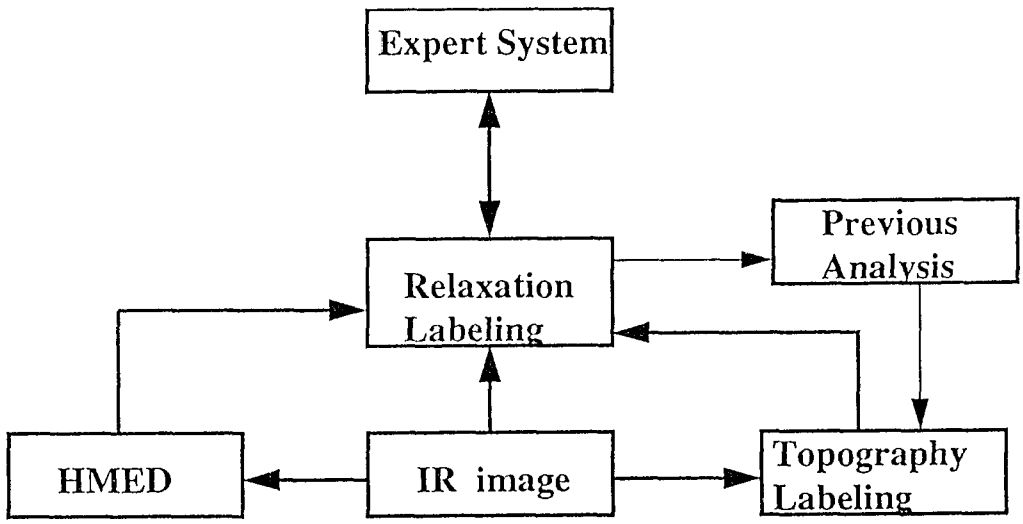


Figure 2: Architecture of SAMAS

Traditionally, to improve the effectiveness of the three modules, a feedback from the medium-level to the low-level module and from the high-level to either medium-level or low-level or both is provided. Tenenbaum and Barrow [52] demonstrated the effectiveness of feedback of domain specific knowledge from the medium to the low-level (the concept of interpretation guided segmentation). Nagao [33] suggests that the segmentation process should be a trial-and-error recursion between the data-driven segmentation and the goal-directed recognition in order to reach a better segmentation of the scene. They also proposed a compact and intimate intergration of segmentation with interpretation, in which an initial segmentation using a standard image-processing procedure is approximately interpreted and repeatedly modified, for interpretation to become complete and consistent. Hanon and Riseman [12] implemented a vision system for natural scenes in which feedback paths were provided to integrate semantics of the domain into segmentation processes, with the notion that segmentation errors will be reduced with the help of domain specific knowledge. We discuss a vision system for interpretation of oceanographic data.

1.6 Edge Detection in Oceanographic Images

The Naval Research Laboratory began development of the Semi-Automated Mesoscale Analysis System (SAMAS), a comprehensive set of algorithms to handle the automated analysis problem, from low-level segmentation through intermediate-level feature formation and higher-level Artificial Intelligence modules that estimate positions of previously detected features when cloud cover obscures direct observation in the current image set [26].

The current version of SAMAS (Figure 2) groups various modules into these three categories. A cloud detection algorithm processes the thermal infrared image of the ocean to classify all pixels either as cloud pixels or non-cloud pixels [51]. Considering only non-cloud pixels, the system uses the cluster shade texture measure as the low-level operation and detection of zero crossings in cluster shade as the medium-level operation, leading to a set of edge primitives [16]. SAMAS uses two-step nonlinear relaxation [20] to label edge primitives [22]. In the first relaxation labeling step, *a priori* probability values of the edge pixels are computed using *a priori* knowledge of the approximate sizes and positions of the features, based on a previous analysis (typically from one week earlier). In the second step, these probability values are updated using compatibility coefficients in an iterative fashion, until the values stabilize. The relaxation labeling technique reduces uncertainty in the assignment of labels to edge pixels [22]. We also developed a topographic-based feature labeling module that uses the surface topology of a pixel and its neighborhood [24, 25].

First we discuss the features of the edge detection algorithm proposed by Holyer and Peckinpaugh, the popular derivative-based edge operators, viz Sobel's operator, are shown to be too sensitive to edge fine-structure and to weak gradients to be useful in this application. The edge algorithm proposed by Holyer and Peckinpaugh is based on the cluster shade texture measure, which is derived from the gray level co-occurrence (GLC) matrix. The authors suggest that the edge detection technique can be used effectively in the automated detection of mesoscale features. The (i,j)th element of the GLC matrix , $P(i,j \mid \Delta x, \Delta y)$, is the relative frequency with which two image elements, separated by distance $(\Delta x, \Delta y)$, occur in the image, one with

intensity level i and the other with intensity level j. The elements of the GLC matrix could be combined in many different ways to give a single numerical value that would be a measure of the edges present in the image. Holyer and Peckinpaugh use a cluster shade function, which is found to be very effective in the edge detection process. Given an M x N neighborhood of the input image containing intensity levels ranging from 0 to (L-1), let f(m,n) be the intensity level of the pixel at sample m, line n of the neighborhood. Then,

$$P(i,j \mid \Delta x, \Delta y) = \sum_{n=1}^{N-\Delta y} \sum_{n=1}^{N-\Delta y} A \tag{1.1}$$

where:

$$A = \begin{cases} \frac{1}{(M-\Delta x)(N-\Delta y)} & if\ f(m,n) = i\ and\ f(m+\Delta x, N+\Delta y) = j \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

The Cluster shade texture measure is then defined by the equation below.

$$S(\Delta x, \Delta y) = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i + j - u_i - u_j)^3$$

where:

$$u_i = \sum_{i=0}^{L-1} i \sum_{j=0}^{L-1} P(i,j \mid \Delta x, \Delta y)$$

$$u_j = \sum_{j=0}^{L-1} i \sum_{i=0}^{L-1} P(i,j \mid \Delta x, \Delta y)$$

The new edge algorithm computes the cluster shade function at each pixel. Then the edges are detected by finding the significant zero crossings in the cluster shade

image. Note that (1) the cluster shade can assume either positive or negative values, (2) the values are the largest in the vicinity of the north wall of the Gulf Stream, (3) values are negative on one side of the wall and positive on the other side, and (4) the transition point from large positive to large negative values coincides with the exact location of the edge of the Gulf Stream. These observations suggest that one could detect edges by finding significant zero-crossings in the cluster shade image.

Significant zero-crossings in the cluster shade image is determined as follows. An initial threshold is selected. Then, for each 3 x 3 pixel neighborhood in the cluster shade image, the absolute value of center pixel is tested with the threshold. If it is less than the threshold, a value "0" assigned in the binary output image. If the absolute value of the center pixel exceeds the threshold, then a test of the neighboring pixels is performed. If the absolute value of any of the eight neighboring pixels also exceeds the threshold, but is opposite in sign from the center pixel, a value "1" is assigned in the binary image indicating the presence of an edge pixel. Because edges are detected by finding zero crossings, precisely positioned lines result, even if the GLC matrix is calculated using a larger window. So, the desired edge detection characteristic of retaining sharp edges while eliminating edge detail is achieved by the new algorithm. It is known that using large windows in derivative-based edge detector algorithms results in poor smoothing. This problem is circumvented in the new algorithm. As an input to our feature labeling algorithm, we used the output image generated by cluster shade algorithm, with a window size of 16x16 pixels and zero crossing threshold of 50. The edge magnitudes obtained from this new edge detector algorithm are used as an input to the feature labeling algorithm. In particular, the edge magnitudes are

Figure 3: Edges extracted from Figure 1 using CSED

used to evaluate the a priori probability values. Figure 3 shows the output of the edge detector on image 1 (Figure 1).

Cayula and Cornillon [4] have developed an edge-detection algorithm for oceanographic satellite images. Their algorithm operates at three levels: picture level, window level, and local/pixel level. At the picture level, most obvious clouds are identified and tagged so that they do not participate at the lower levels. The cloud-finding procedure is based on temperature and shape. At the window level, the temperature distribution in each window is analyzed to determine the statistical relevance of each possible fronts, using unsupervised learning techniques. Finally, local edge operators are used to complete the contours found by the region-based algorithm. Since the local operations are used along with the window-based algorithm, the qualities of scale invariance and adaptivity associated with the region-based approach are not lost. Cayula [3] compare the edge-detection algorithm described above. The main difference is that the algorithm given by Holyer and Peckinpaugh operates at the local level only, while that by Cayula and Cornillon is a multilevel algorithm. Refer to [3] for the details on the results of the comparison.

Chapter 2

Morphological Edge Detectors

Mathematical morphology based on geometric shape, is used in biomedical image processing, robot vision systems, and low-level vision problems for its conceptual simplicity. Many techniques in computer vision use mathematical morphology as a tool for the extraction of features and recognition of objects. Matheron [29] introduced the application of mathematical morphology for analyzing the geometric structure of metallic and geologic samples. Serra [47] applied mathematical morphology for image analysis. Haralick [15] presented a review of mathematical morphology applied to image analysis.

Pelag and Rosenfeld [38] use gray scale morphology to generalize the medial axis transform to gray scale imaging. Pelag *et al* [37] measure changes in texture properties as a function of resolution using gray scale morphology. Werman and Pelag [56] use gray scale morphology for texture feature extraction. We will study the use of gray scale morphology and texture information for edge detection in oceanographic images.

Recently, mathematical morphology has been applied for the extraction of edges. Most of the template based edge detectors are known to perform satisfactorily under high signal to noise ratio, but degrade significantly when noise is introduced into the system. Some of the template based edge detectors are the Prewitt operators and the Kirsh operator. A number of edge detectors fit a polynomial function on the image data. Then, the first and second directional derivatives are computed, from which the edges are extracted. Mathematical morphology based edge detectors have been shown to out-perform most spatial and differentiation based edge detectors [45]. Morphological edge detectors are local neighborhood nonlinear operators. Morphological techniques tend to simplify image data while preserving the shape characteristics and eliminating irrelevancies. These algorithms often generate useful and surprising results. We briefly present preliminary concepts of mathematical morphology. Matheron [29] gives a detailed discussion of mathematical morphology.

2.1 Preliminary Concepts

An image "f" is a set of pixels in a rectangular array (mesh). $f(i, j)$ is a pixel at coordinate (i, j) in the image f. A structuring element is analogous to the kernel/template of a convolution operation, and it is associated with a pre-designed shape. A structuring element may have any shape. Morphologic operators can be visualized as working with two images, the original image and the structuring element. The structuring element is used as a tool to manipulate the image using various operations namely *dilation, erosion, opening, closing.* The dilation of a binary image f by a

structuring element S is defined as

$$f \oplus S = \{ a + b \mid a \in f \wedge b \in S \}$$

The erosion of a binary image f by a structuring element S is defined as

$$f \ominus S = \{ a - b \mid a \in f \wedge b \in S \}$$

The "*dilation*" d of a gray-scale image f by a structuring element S is defined as

$$d(i,j) = MAX(f(i+x, j+y) \oplus S(x,y))$$

where x and y are the coordinates of a cell in S whose center cell is the origin, and (i+x, j+y) is in the domain of f. Similarly, "erosion" of a gray-scale image f by a structuring element S is defined as

$$e(i,j) = MIN(f(i+x, j+y) \ominus S(x,y))$$

The *closing* operation is a dilation followed by an erosion, and similarly *opening* is an erosion followed by a dilation. Thus, closing is defined as

$$c(i,j) = MIN(d(i+x, j+y) \ominus S(x,y))$$

where d is the dilated image of original image f. Opening is defined as

$$o(i,j) = MAX(e(i+x, j+y) \oplus S(x,y))$$

where e is the eroded image of original image f. A sequence of these gray scale morphological operations on an image often produces useful results. For instance, a simple morphological edge detector is the *dilation residual* edge image, defined as

$$DR\ (i\ ,\ j) =\ d\ (i\ ,\ j)\ -\ f\ (i\ ,\ j)$$

Similarly, the *erosion residual* edge detector is given by

$$ER\ (i\ ,\ j) =\ f\ (i\ ,\ j)\ -\ e\ (i\ ,\ j)$$

Even though these edge detectors are simple and robust, they are not reliable for extremely noisy images, and introduce spurious edges.

2.2 Blur-Minimization Morphological Operator

Lee *et al* [17] designed a Blur-Minimization Morphological (BMM) operator for edge detection. The BMM operator blurs the original image by averaging the pixel values spanned by the structuring element. Dilated and eroded images are generated from the blurred image. Dilation residual and eroded residual images are created using these images. The edge strength at coordinate (i, j) is given by the minimum of the dilation residual and erosion residual. Symbolically we write

$$\mathrm{BMM(i,\ j)} = \mathrm{MIN}\ (\ f_a\mathrm{(i,\ j)} - \mathrm{e(i,\ j)},\ \mathrm{d(i,\ j)} - f_a\mathrm{(i,\ j)})$$

where $f_a = \frac{\sum f(i+x,j+y)}{N}$ is the blurred image, N is the number of cells in the structuring element, and (i+x, j+y) is defined in the domain of the image. In spite of being conceptually simple and computationally efficient, the BMM edge detector has been proven to perform better than the spatial and differential based edge detectors.

2.3 Alpha-Trimmed Morphological Operator

Feehs and Arce [42] showed the importance of blurring the original image for morphological edge detection. They introduced a *Alpha-Trimmed multidimensional Morphological* (ATM) edge detector that incorporates the opening and closing operations. They also proved statistically that ATM performs better than BMM. Let us consider the ATM edge detector for 2-dimensional images with a structuring element of size n x n. The original image is initially blurred by $f_a = \frac{\sum_{i=\alpha+1}^{k-\alpha} f_i}{k-2*\alpha}$ where $k = n^2$ is the number of pixels in the original image spanned by the structuring image, f_i is the i^{th} smallest valued pixel in the sorted sequence of pixels in f spanned by the structuring element, and α is the trimming factor. If α is 0, we consider all pixels spanned by the structuring element for blurring. If $\alpha = i$, we consider all sorted pixels greater than f_i and less than f_{k-i} spanned by the structuring element. The edge strength at (i, j) computed by the ATM edge detector is

$$\text{ATM(i, j)} = \text{MIN } (\ (\ o\ (i,j) - e\ (i, j)),\ d\ (i, j) - c\ (i,j)\ )$$

in which the erosion and dilation operation are performed on the α trimmed blurred image, and the opening and closing operation are performed on the eroded and dilated images of the α trimmed blurred image.

The ATM edge detector, like the BMM edge detector, is unable to extract the weak gradients associated with certain mesoscale features [25]. This could be possibly because the definition of gray-scale dilation and erosion considers only the maximum and the minimum intensity pixels in a given neighborhood of a pixel. As a result, the dilation and erosion residual values are not sufficient for these edge detectors to

pick up the weak gradients. For increased structuring element sizes, weak gradients are extracted along with other spurious edge pixels which are difficult to isolate.

The cluster shade algorithm [16] presented earlier extracts most of the weak gradient valued pixels along with the strong gradient valued pixels. This is due to the application of a texture based algorithm in an application where multiple gradient values are vital for interpretation. The algorithm is very computation intensive [16].

We seek a low-level segmentation module that is simple in design and construction, despite making use of the texture information in the image. We anticipate that such a design would extract all the boundaries of the features irrespective of their gradient values. One of the possible methods of making use of texture information is to compute the first order histogram in a neighborhood of a pixel.

2.4 Motivation and Scope

Previous morphological edge detectors are designed to work only in the image domain. Such designs ignore the vital information contained in the histogram of an (sub)image. As a consequence, various weak gradient values pertaining to important features are missed in oceanographic IR images. We expect that a morphological edge detector that incorporates information from the image histogram will provide improved performance while being conceptually simple and computationally efficient. we propose new morphological operations defined over the histogram of a neighborhood of a pixel. The new morphological operations are limited to erosion and dilation

only, and the morphological basis of these new operations is explained in the context of oceanographic images only [23].

Chapter 3

Histogram-Based Morphological Edge Detector

The histogram is a popular tool used in image processing and image analysis. It is used for edge detection, thresholding, texture feature extraction and other related problems. Let H be the histogram of an image or sub-image, let $g_0, g_1, ..., g_{l-1}$ be the gray levels for which the histogram is defined, and $h(g_0), h(g_1), ..., h(g_{l-1})$ be the count values for those gray levels. Previously, researchers have designed image segmentation methods from the histogram using either global or local thresholding concepts. For instance, when a light object is present in dark background, the histogram may have twin peaks occurring at the intensities corresponding to the intensities of the object and background. A suitable threshold between the two peaks is selected to segment the object from the background [49]. When multiple objects are present in the background, a global histogram is of little use. However, a local histogram in the neighborhood of a pixel would exhibit twin peaks from which an object can be

the neighborhood of a pixel would exhibit twin peaks from which an object can be segmented from the background [45].

It is noted that the gray scale dilation and erosion are the maximum and minimum of the image pixels spanned by the structuring element, respectively. The definitions of gray scale morphology, in fact, make use of the histogram indirectly. This is explained using a structuring element S of height 0 in the following way: gray scale dilation over the histogram is the maximum of $g_0, g_1, g_i, ...g_{l-1}$ for which $h(g_i) \neq 0$. Similarly, the gray scale erosion is the minimum of $g_0, g_1, g_i, ...g_{l-1}$ for which $h(g_i) \neq 0$. The average of the image pixels is computed from the histogram. It is also noted that the BMM and ATM edge detectors extract edges using the gray scale dilation and erosion operations. But, these definitions consider only the maximum and minimum of the image pixel intensities in a given neighborhood. Thus, we infer that there is a clear distinction in theories between the histogram based edge detectors and morphology based edge detectors. The essential ideas in the former methods stem from the fact that the histogram taken near the boundaries exhibits twin peaks, while the latter methods mark a pixel as an edge pixel depending on the maximum and minimum intensity values of the pixels near the boundaries. We anticipate that morphological edge detectors that use the histogram in an effective way would reduce the gap between these two edge detection theories. In doing so, we will develop extensions to the definitions of morphological operations in the domain of the histogram, but not in the domain of the image. We anticipate that such extensions provide us new directions in the notion of morphology-based edge detectors, particularly in the context of oceanographic images.

Let a histogram H defined over gray levels $g_0, g_1, ..., g_{l-1}$, be computed using the image pixels spanned by the structuring element S centered at the coordinates (x, y). g_0 and g_{l-1} are the intensity of black and white pixels, resp. Call the height of the histogram at these gray levels $h(g_0), h(g_1), ..., h(g_{l-1})$. Let the intensity of the pixel (at coordinates (x, y)) - where the histogram is computed - be g_i. We define histogrammic dilation *h-dilation* at a pixel (x, y) as

$$d_h(\text{x, y}) = \{g_j \mid h(g_j) = \max[h(g_i), h(g_{i+2}), ..., h(g_{l-1})] \text{ and } (\text{i} \leq \text{j} \leq \text{l-1})\}$$

Similarly, we define the histogrammic erosion *h-erosion* as

$$e_h(\text{x, y}) = \{g_j \mid h(g_j) = \max[h(g_0), h(g_1), ..., h(g_i)] \text{ and } (0 \leq \text{j} \leq \text{i})\}$$

It is noted that both the d_h and e_h are defined in terms of peaks of the histogram on either side of the gray level intensity g_i of the pixel. By defining these operations in this fashion, we make a noticeable deviation from the traditional dilation and erosion operations.

The value of e_h is the gray level intensity g_e at which the histogram height is the maximum of all heights computed at gray level intensities lower than the (average) gray level intensity of the pixel. The value of d_h is the intensity g_d at which the histogram height is the maximum of all histogram heights computed at intensities greater than the (average) intensity of the pixel. In case an unique intensity g_e (g_d) is not found, g_e (g_d) that is closer to g_i is selected.

One of the motivations for using h-dilation and h-erosion is as follows. Figure 1 is unusually free of clouds. Even though many cloud detection algorithms are

available, none of them detect all cloud pixels [51]. Therefore some cloud pixels will be present in the input image. We recall that the traditional dilation and erosion definitions consider only the extreme values in the neighborhood of a pixel. If a cloud pixel is one of the extreme values, then an edge detector based on traditional mathematical morphology will extract spurious edge pixels. We anticipate a reduction in the extraction of spurious edge pixels when we use the h-dilation and h-erosion operations.

A careful examination of these definitions indicates a strong link between the histogram based and morphology based edge detectors. For instance, consider a histogram computed in a neighborhood of a pixel near the boundary having (twin) peaks with the average intensity falling between the peaks. The histogram based methods search for the *valleys and peaks* in the histogram, whereas the morphological methods (BMM and ATM) search for the *extreme* intensities that have non-zero histogram heights.

Figure 4 is the new edge detector that makes use of the histogrammic dilation and erosion. The edge strength in the edge image at (x, y) is given by

$$\mathrm{HMED}(x, y) = \min (f(x, y) - e_h(x, y), d_h(x, y) - f(x, y))$$

where f (x, y) is the average intensity g_i at coordinate (x, y).

Thus, the edge magnitude values in the output image are computed in a similar fashion to that of the BMM and ATM edge detectors. Normally, an edge detector should generate edge pixels of width two in case of ideal step edges. However, this is not true for edge detectors that take input images which are blurred or smoothed ver-

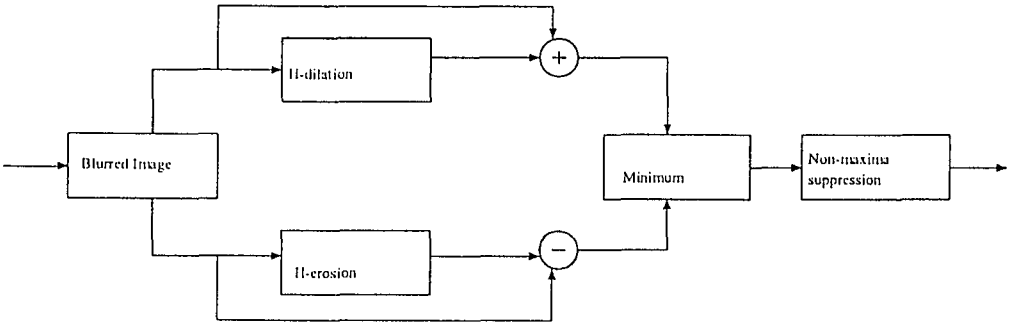


Figure 4: Architecture of HMED

sions of the original image. The HMED technique, when used with large structuring elements, produces significant non-zero edge strength of width more that one pixels. This is consistent with the fact that HMED blurs the original image. Usually, the true edge pixels get assigned higher edge strength than their neighbors. Normally, a suitable threshold is selected to extract the true edge pixels. However, HMED extracts the true edge pixels using a non-maxima suppression technique.

Thus, we establish a computational framework that adapts the advantages of two edge detection theories. With such a computational framework, we show that HMED performs better than the BMM and ATM edge detectors.

3.1 HMED Algorithm

The HMED algorithm given below takes a blurred image as input and produces an edge strength image in which non-maxima suppression has to be performed. This procedure uses functions GET_INTENSITY and HIST(i) that return the intensity g when the coordinates (x, y) of a pixel are given, and the count of pixels with gray level intensity i, resp. The constant MAXGRAY = 255 is defined.

It is noted that a new histogram is not computed from scratch at every pixel's neighborhood. The histogram of the adjacent neighborhood (x, y+1) is computed by using the histogram computed at pixel (x, y) as described in version 6 in [12].

The HMED algorithm presented here involves only parameters such as the size of the structuring element and non-maxima suppression. No strict rules can be stated in this regard. For the images considered here we present results that are produced

```
PROCEDURE HMED(blur_img, dil_img, erd_img, out_img, W)
BEGIN
1.  for each pixel P(x, y) in the blurred image
        BEGIN
2.          compute a histogram H of the sub-image (W × W) centered at (x,y)
3.              max = get_intensity (x, y)
4.              For i = max to MAXGRAY (* dilation operation *)
5.                  if ( hist(i) > hist(max) )
6.                      max = i;
7.                  dil_img(x, y) = max ;
8.              max = get_intensity (x, y)
9.              For i = 0 to max (* erosion operation *)
10.                 if ( hist(i) > hist(max) )
11.                     max = i;
12.                 erd_img(x, y) = max ;
13.         temp1 = blur_img(x,y) - erd_img (x, y);
14.         temp2 = dil_img (x, y) - blur_img (x, y);
15.         out_img (x, y) = MIN ( temp1, temp2);
        END
END
```

with structuring element sizes 15 x 15, 17 x 17, 19 x 19 and 21 x 21.

There are at least two ways in which we can extract the true edges: computation of zero crossings and suppression of non-maxima.

1. to extract zero-crossings, let us consider line 15 in the algorithm. If temp1 is minimum, then -temp1 (negative value) is assigned in the edge strength image, otherwise +temp2 (positive value) is assigned. Then a zero-crossing test has to be performed. The significance of a negative value is that the histogram is skewed to the negative side of the mean value. A positive value indicates that the histogram is skewed to the positive side of the mean value.

2. In case of non-maxima suppression in the edge strength image, we suppress a pixel as a non-edge pixel if there exists a group of pixels whose value is much greater than the pixel to be suppressed [24].

3.2 Handling of Cloud Cover

The test image in Figure 1 is unusually free of clouds. A typical oceanographic image contains cloud cover as well as attenuation due to water vapor. Thus the low-level vision algorithms have to be designed to handle the cloud cover. One simple method to avoid the cloud pixels is to generate a cloud mask using a technique proposed in [6]. The cloud mask is a binary image that contains the values 0 or 1. A value 0 signifies that the pixel is part of a cloud and 1 signifies a non-cloud pixel. In this application, cloud pixels are treated as follows:

1. A pixel in the IR image is considered a candidate edge pixel if and only if the cloud mask has value 1 at the same coordinate.

2. For a candidate edge pixel, the histogram is computed by considering only the non-cloud pixels

3.3 Implementation Results

The test data set consists of 12 satellite IR images of the North Atlantic. We present results from 3. We processed the images with the BMM and ATM detectors using structuring elements of sizes 5 x 5, 7 x 7, 9 x 9, 11 x 11, and 13 x 13, while we used structuring element sizes of 13 x 13, 15 x 15, 17 x 17, 19 x 19, and 21 x 21 with the HMED detector. The ATM edge detector's α parameter was 3 in all cases. We do not know of any strict rules to govern the choice of the structuring element's size or the value of α.

Figures 5 and 6 all show the results obtained with Figure 1. Figure 5 shows the results of applying the BMM edge detector with structuring elements of sizes 5 x 5, 7 x 7, 11 x 11, and 13 x 13. Increasing the structuring element's size results in finding more edges. Figure 6 shows the results of applying the ATM detector with structuring elements of sizes 5 x 5, 9 x 9 , 11 x 11, and 13 x 13. The results are slightly different, but again increasing the structuring element's size finds more edges. Figure 7 and Figure 8 are the h-dilated and h-eroded images of Figure 1. Figure 9 is the gradient image created by taking the minimum of the residuals (line 15 of the algorithm). Figure 10 shows the results of applying the HMED detector with window sizes 13 x

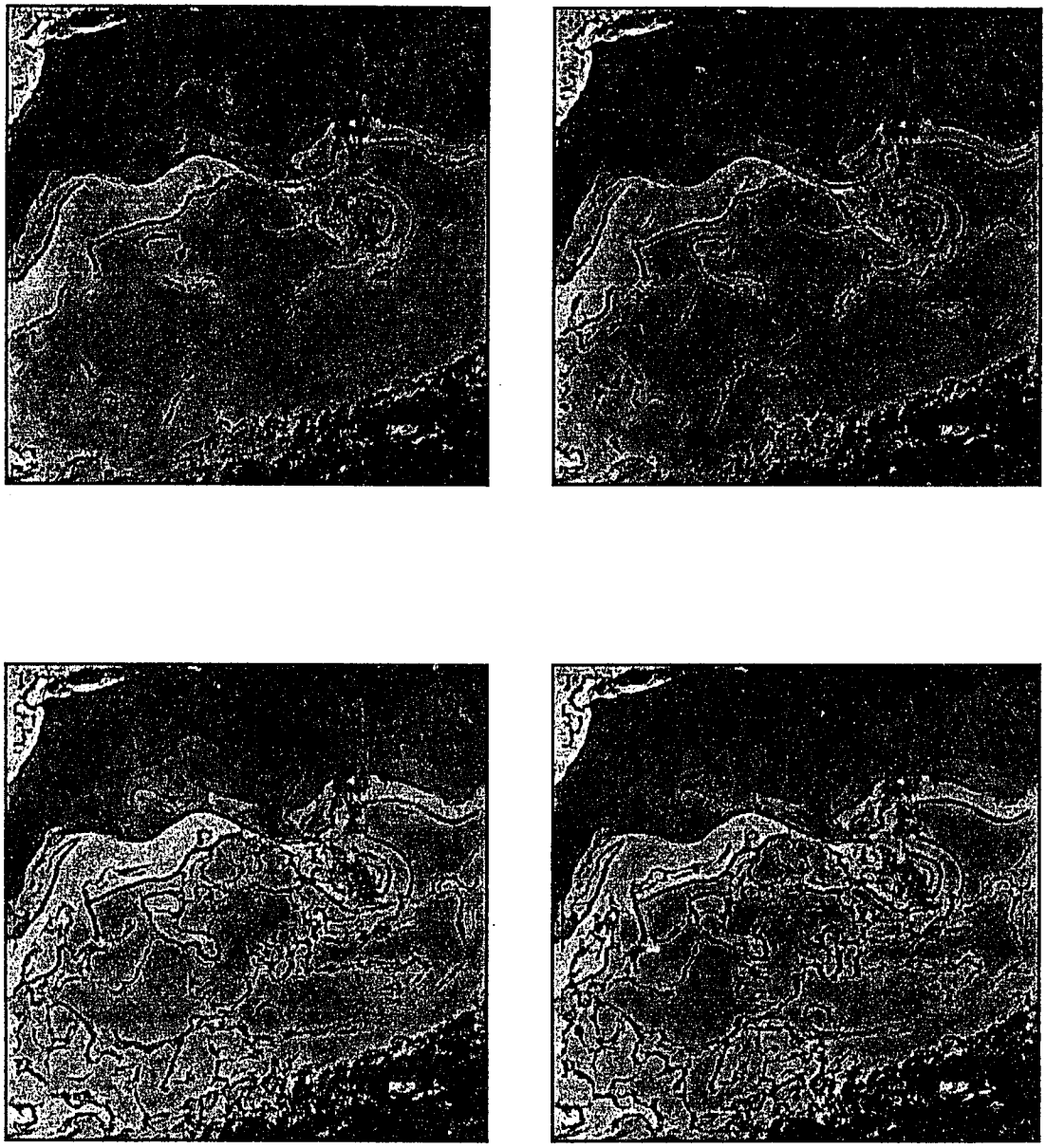


Figure 5: Results of applying BMM on Figure 1, 5x5 (top left),
7x7 (top right), 11x11 (bottom left), 13x13 (bottom right)

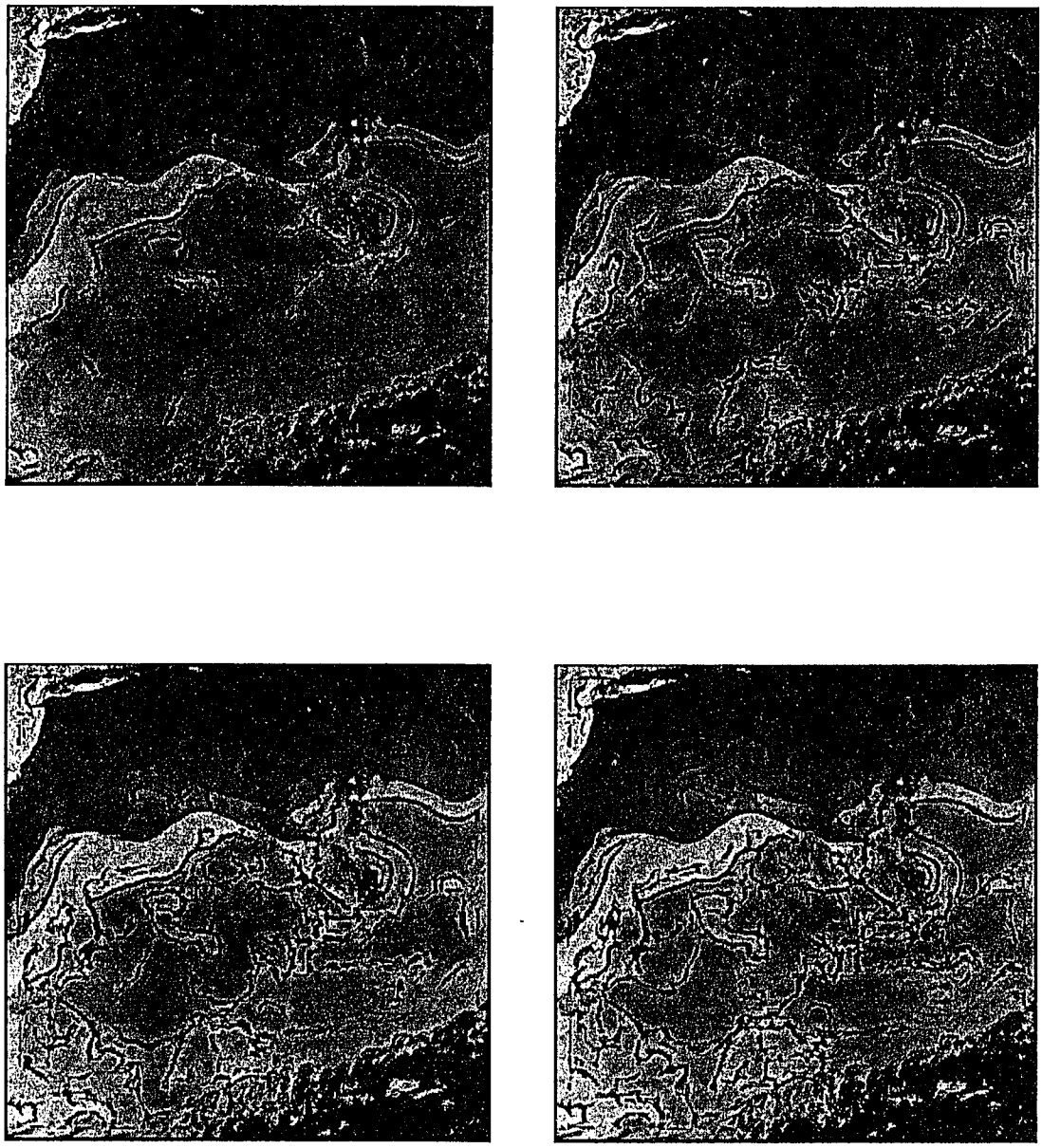


Figure 6: Results of applying ATM on Figure 1, 5x5 (top left),
9x9 (top right), 11x11 (bottom left), 13x13 (bottom right)

Figure 7: H-Dilated image of Figure 1

Figure 8: H-Erroded image of Figure 1

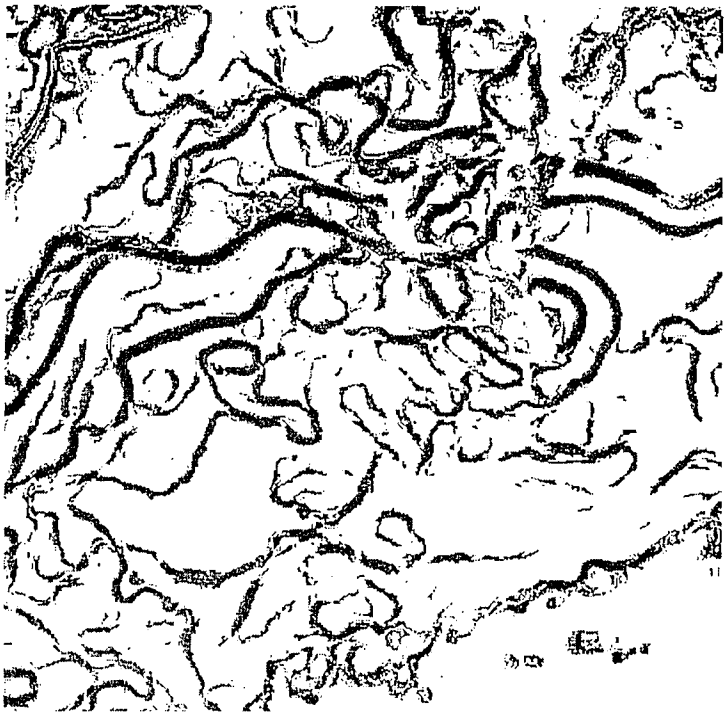


Figure 9: Gradient image using Figure 7 and 8

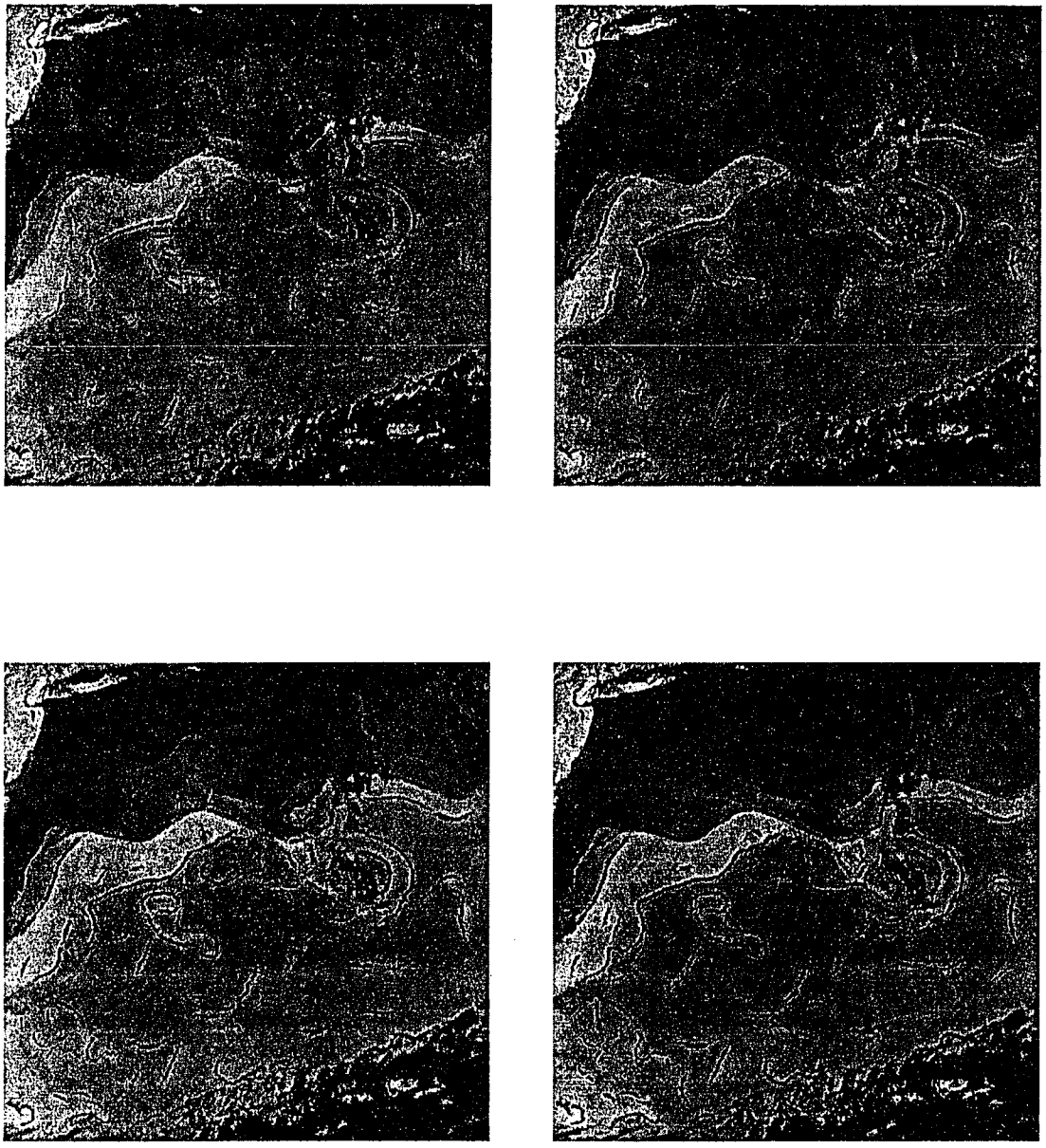


Figure 10: Applying HMED on Figure 1 - 13x13 (top left),
15x15 (top right), 17x17 (bottom left), 19x19 (bottom right)

Figure 11: North Atlantic image obtained on April 10

Figure 12: North Atlantic image obtained on April 21

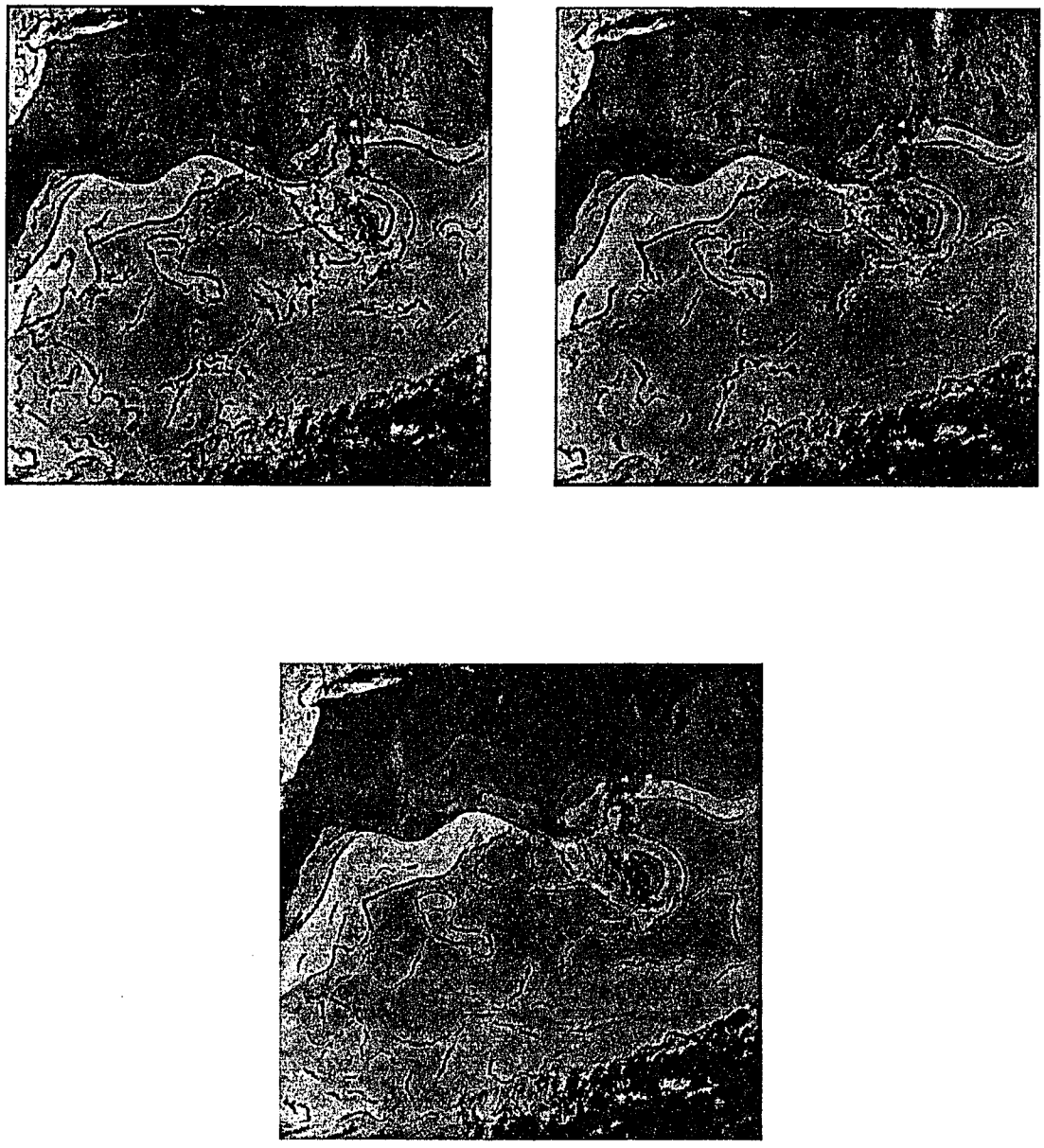


Figure 13: BMM with 9x9 (top left), ATM with 7 x 7 (top right) and HMED with 21x21 (bottom left) all using Figure 1

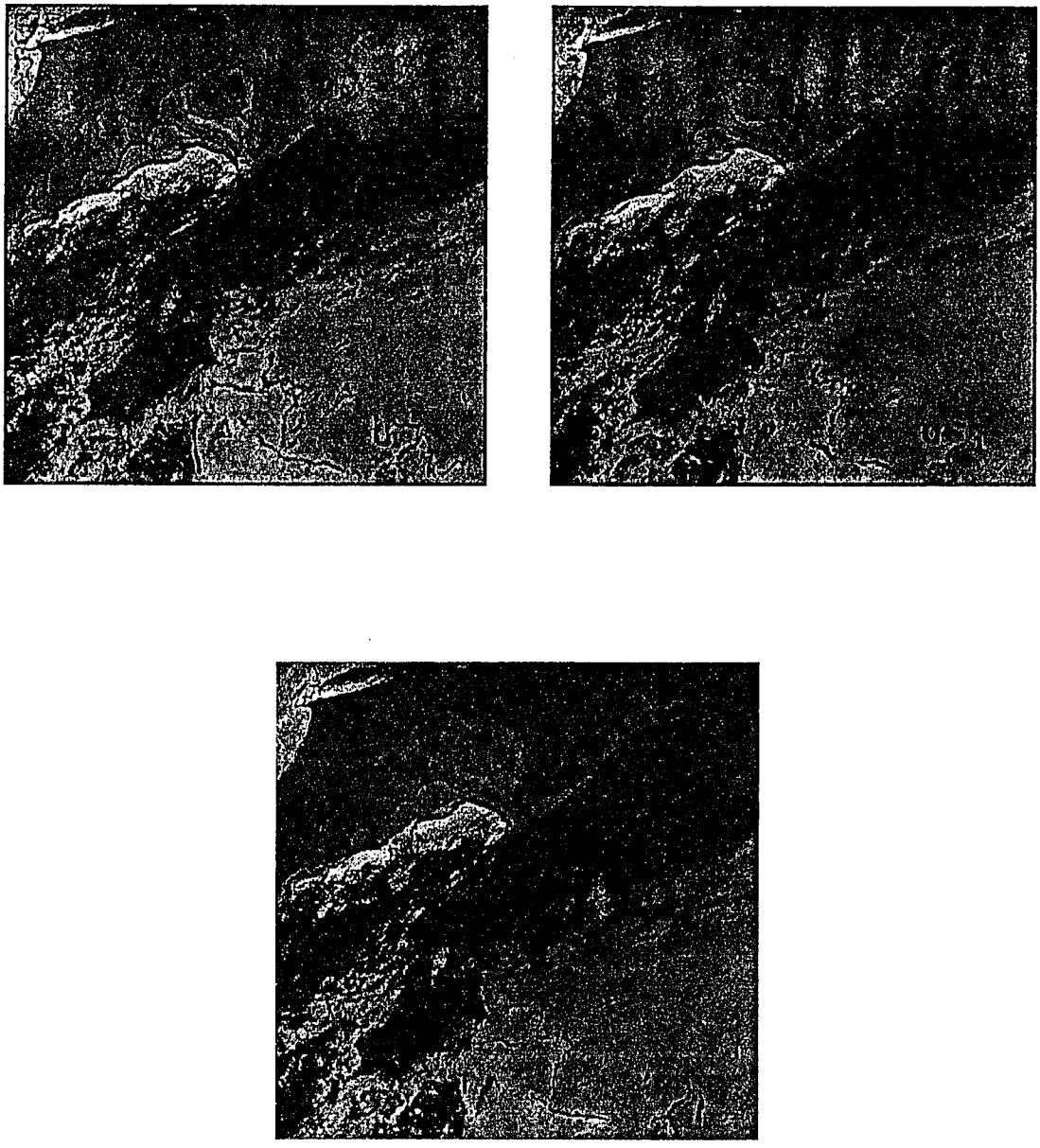


Figure 14: BMM with 9x9 (top left), ATM with 7 x 7 (top right) and HMED with 21x21 (bottom) all using Figure 11

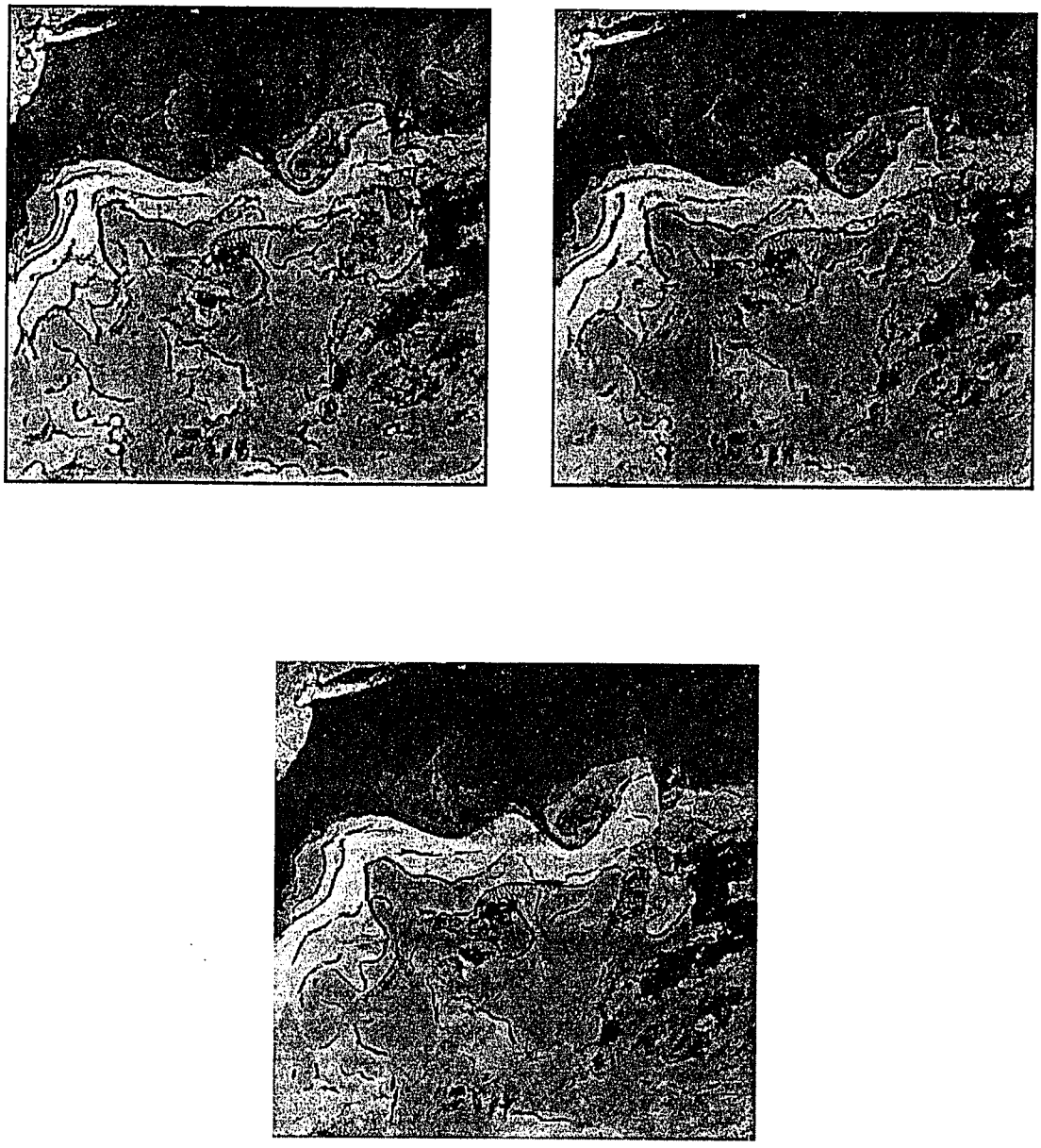


Figure 15: BMM with 9x9 (top left), ATM with 7 x 7 (top right) and HMED with 21x21 (bottom left) all using Figure 12

13, 15 x 15, 17 x 17, and 19 x 19. The HMED detector finds far fewer spurious edges, and increasing the window size seems to increase the continuity of the edges without finding many more.

Figures 11, and 12 are also satellite images of the North Atlantic. Figures 13, 14, and 15 show a comparative analysis of the three methods applied to the three images. In each case, the results with the structuring element judged to give the best performance is shown. All three methods find the Gulf Stream's North Wall and boundaries of warm eddies, for all structuring elements. These edges have high gradient values, so detection is relatively easy. The South Wall and boundaries of cold eddies are spatially distributed over 6-7 pixels with low gradient values. None of the detectors do as well in extracting these weak gradients as in finding the stronger ones, when using small structuring elements. Increasing the size causes the BMM and ATM detectors to introduce many spurious edges. However, the HMED detector is able to extract these weak gradient values without introducing many spurious edge pixels.

Again, HMED is able to extract the boundaries of the mesoscale features without introducing spurious edge pixels. We conclude that the HMED's better performance is due to the use of h-dilation and h-erosion.

Table 1. Computing Time of HMED on SUN-SPARC	
WINDOW SIZE	TIME in Seconds
13 x 13	150 seconds
15 x 15	152 seconds
17 x 17	156 seconds
19 x 19	157 seconds
21 x 21	163 seconds

3.4 Parallel HMED Implementation

Parallel computing for image processing and computer vision has received considerable attention during the last few years. The primary reason is due to several tasks in image processing that needs to be implemented in either parallel or distributed fashion. For instance, the learning phase in neural networks takes hours to days to create the knowledge base. The relaxation technique converges after couple of hours. The Hough Transform detects circles and ellipses after using the computer resources for

several minutes. The fourier transform involving huge matrices takes several minutes. The above mentioned methods not only require prolonged use of the CPU, but also require huge amount of computer memory to store the image. Techniques have been developed for sequential processors to reduce both the memory requirements and the computational complexity. However, further improvements can be made using parallel computers. While a large number of parallel computer architectures are designed, very few have been built and tested with real applications. The two main exceptions to this are mesh connected processor arrays and hypercube multicomputers. Both of these schemes are commercially implemented by a number of vendors, and tested by a large number of researchers for real applications.

Mesh connected computers are of particular interest to the image processing community because the topology of the pixels in the image and the processors in MCC matches perfectly and the nearest-neighbor interconnection among processors preserves the spatial relation among image pixels while maintaining a low-hardware implementation cost. Moreover, mesh-connected arrays can efficiently handle low level-image processing applications which require only local operations on image pixels.

All Processing Elements (PEs) in MCC are arranged in a 2D rectangular grid of size M x M where each PE is directly connected to four-neighbors, each one in North, East, South and West direction. Thus, a PE can easily communicate to its four neighbors. This feature is extremely suited to image processing operations such as convolution, template based edge detection in which each pixel needs the pixel value of its four neighbors. Each PE maintains a local memory for storing data and receives

instructions from a centralized controller. All PEs work in a synchronized fashion under a Single Instruction, Multiple Data (SIMD) Stream mode. This means all the active PEs execute the same instruction, but on different data. This mode of operation is needed in image processing algorithms because the algorithms predominantly perform the same instructions on different data sets.

3.4.1 Motivation for Parallel Processing

The motivation of designing a parallel HMED algorithm is primarily due to the sequential HMED algorithm processes a 512x512 image for almost 150 seconds on a SUN-SPARC workstation, and this is acceptable if only one image is to be processed. The Cluster-Shade algorithm discussed before runs for almost 10 minutes for the same 512 x 512 image. However, this is not acceptable when a large number of images are to be processed. This led us to investigate a possible reduction in the running time of the HMED algorithm on a parallel machine, such as MASPAR, whose architecture is similar to the MCC. We provide a brief description of MASPAR and the implementation of HMED on MASPAR.

MASPAR consists of 128 x 64 processors arranged in a 2 dimensional grid - with 128 processors horizontally and 64 processors vertically. The processors operate in a SIMD fashion with the instructions provided by a centralized processor (controller). The controller is a unix-based machine and the parallel language is called MPL which is an extension of 'C' language. Each processor has an *iproc* to uniquely identify the processor. A processor can be also be identified by a pair of numbers *ixproc, iyproc* where *ixproc* is the x-coordinate and *iyproc* is the y coordinate of the processor. The

top-left processor's *iproc* is 0 and *ixproc, iyproc* pair is (0, 0). Similarly, the bottom-right processor's *iproc* is 8192 and *ixproc, iyproc* pair is (127, 63). Each processor has a local memory of 128,000 bytes.

A processor can be enabled or disabled to participate in an operation by setting a mask. The 'for' statement given below, for instance,

```
if (iprocis not 25)
Begin
        .
        do the loop statements
        .
End
```

is executed by all processors except processor 25 or (0, 25).

```
if (ixprocis 45)
Begin
    .
        do the loop statements
    .
End
```

is executed by all processors in column 45.

The next step involves the mapping of 512 x 512 image onto the 128 x 64 array of processors in MASPAR. The main factors in mapping are : size of the sub-images to be mapped and the window size to be used by HMED. HMED uses window sizes 15 x 15, 17 x 17, 19 x 19 or 21 x 21. HMED computes the gradient value of a pixel *P(x,y)* using the pixels values of pixels that are present in a window centered at (x,y) coordinates. We consider a few of them and discuss their advantages and disadvantages.

We implemented a simple scheme **read-block method** in which each processor is allowed to read block of pixels (or sub-image) directly from the image. The method partitions the image into a number of sub-images and assigns each sub-image to a processor. Each processor implements the HMED on the sub-image just like the sequential technique. However, the border pixels cannot be processed because of some missing pixels that are present in the adjacent processors. For example, let us consider a 32 x 32 sub-image and window size 15 x 15. A processor processes the sub-image in middle region of size 17 x 17, but not the seven rows of pixels on top and bottom and seven columns of pixels on left and right of the sub-image. Thus processor exchanges the seven rows of top pixels for the bottom seven rows of pixels in the north adjacent processor. The bottom seven rows of pixels in the sub-image are exchanged for the top seven rows of pixels in the south adjacent processor's sub-image. The left seven columns of pixels are exchanged for the right seven rows of pixels in the west adjacent processor's sub-image. Similar exchange of pixels is done with the east adjacent processor.

In the implementation we used only 64 x 64 processors even though MASPAR at the Concurrent Computing Lab, LSU., has 128 x 64 processors, resulting in a 8 x 8 sub-images in each PE. We tested the computing time for 512 x 512 image using 64 x 64 processors for various window size.

Table 2. Computing Time of HMED on 64 x 64 MASPAR

WINDOW SIZE	TIME in Seconds
13 x 13	6.7
15 x 15	7.2
17 x 17	7.8
19 x 19	8.6
21 x 21	9.2

3.5 Distributed HMED

With ongoing recent advances in high-speed networking and computer technology, distributed systems provide a mechanism for effectively utilizing the total compu-

tational power of multiple workstations available on a local network. We notice, however, massively parallel computers and hundreds of high-performance workstations in our computing environments frequently sit idle for many hours in a day. This led us to investigate ways of distributing the data and the time-consuming image processing tasks to the network of workstations available to us. Unlike a massively parallel supercomputer dedicating uniform and intensive computing power, a network computing environment provides non-dedicated and scattered computing cycles. The primary purpose of this implementation is to document a pilot study using network of machines, connected by Ethernet, to coordinate distributed image processing tasks. Thus, we anticipate that using loosely coupled network of high performance workstations would reduce the total time linearly and process images that are too large to process in a single machine. There exists many computer softwares for coordinating distributed image processing tasks, such as Parallel Virtual Machines **PVM**, to implement the data communications between the machines. PVM lets users write portable application programs to execute on various workstations. To run a program using PVM, daemon process have to initiated on the host and remote machines, and the application program have be loaded on the remote machines. The daemon processes guarantees reliable delivery of data between the remote and host machines. The daemon processes also guarantees communication and coordination between the machines.

In this pilot study, however, we use TCP-IP and UNIX-BSD sockets as a medium of communication and coordination between the remote machines and the host. The communication and coordination is based on the client-server model. As a result

we have to code the basic routines for communication and coordination between the host and remote machines. The host machine, as a server, manages the images to be processed and waits for the clients (remote machines) to request the data or sub-images to process. Soon after receiving a request from a client, the server creates another process as a child process and instructs the child process to transfer the data to the client and receive the processed data from the client. The child process and the client communicate through sockets created.

The server creates shared memory to store input and output images. All child processes can access the images on a mutual exclusion (one by one) basis.

At the creation of the child process, the parent process (server) assigns next un-processed sub-image to the child processes for transferring the data to the client. The server itself can transfer of data, but it has to wait for requests from other clients. The child process knows the exact location of the sub-image in the actual input image. After receiving the data from the child process of the host server, the client execute the HMED algorithm on the input data. After processing the data with HMED, the client sends the processed data back to the host machine (server).

The server program is installed in Silicon Graphics machine at Robotics Research Lab, LSU. The clients (approximately 16) are spread across the entire LSU computing services. We tested the software both during day and night times. The computing time taken by the clients varied from 18 to 30 seconds. It is noted this time is primarily due to the data communication using TCP-IP.

Table 3. Computing Time of HMED on Network of Workstations	
No of Processors	TIME in Seconds
16	18-30 seconds

Chapter 4

Feature Labeling Techniques

4.1 Introduction

Several previous studies have addressed the automation of analysis of IR imagery for mesoscale features. Gerson and Gaborski [8] and Gerson *et al* [9] investigate the detection of Gulf Stream in IR images from the Geostationary Operational Environmental Satellite [GOES]. The resolution of GOES is 8 km/pixel, while AVHRR provides 1.1 km/pixel at nadir. Thus GEOS provides a coarser representation of the ocean surface. Gerson and Gaborski use a hierarchical approach in which 16x16 pixel frames within a image are evaluated for the presence of a Gulf Stream. Frames flagged as Gulf Stream possibilities are further evaluated using a 5 x 5 pixel window within each frames. The evaluation is carried out using mean, standard deviation and second order gray level statistics.

Coulter [6] performed automated feature extraction studies using the higher reso-

lution (1 km) AVHRR data. Mean, standard deviation, and gradient in a 3 x 3 pixel window were combined with em a priori probabilities based on a large data set in order to classify each pixel according to bayes's decision theory. Boundaries between water classes then became an indication of edge locations within an image. Promising results were reported for this method for locating the Gulf Stream. However, Coulter [6] reported that eddy classification is more difficult because their historical statistics are less stationary. Indeed, the requirement for em a priori knowledge and stationary statistics is a limiting factor in the use of this method.

Janowitz [18] studied the automatic decision of the Gulf Stream eddies using AVHRR data. Recognizing that edges in this high-resolution IR imagery are too noisy, i.e., have too much fine-structure detail, for effectively locating eddies, Janowitz started with filtering to smooth the image. Smoothing was followed by an image simplification algorithm [18]. The smoothed and simplified image was then transformed into a binary edge image by a Kirsch edge detector (as described in [41]. The last step was ellipse detection on the binary edge image as a means of locating eddies. The centers of all thess cold eddies in the test image were correctly identified and no edy false alarms were reported. The Janowitz study lends further support to the feasibility of automated mesoscale feature detection.

Nichol [34] uses a Region Adjacency Graph to define spatial relationships between elementary connected regions of constant grey level called atoms. Eddy-like structure is then identified by searching the graph for isolated atoms of high temperature that are enclosed by atoms of lower temperature (for the case of warm eddies). Although satisfactory emulation of human extraction of eddy structure is claimed for

this method, Nichol [34] does not point out that not all enclosed uniform areas identified by this method will correspond to real ocean structure. We agree with this conclusion. It seems possible that relatively uniform areas such as the Sargasso Sea that might be devoid of eddies could possibly contain more uniform areas than the region near the Gulf Stream where eddies exist but the spatial patterns of the sea surface temperature are very complex. The "false alarm" statistics of Nichol's method should be examined.

Although these investigators have reported some success in automating the interpretation process for oceanographic images, they were trying to do the entire interpretation step using conventional image processing techniques.. In other words, there was no sharp division between the non-semantic based modules and semantic based modules, partially because the knowledge-based methodology was not available to provide the tools (Argialas and Harlow, 1990). Because of the growing need for the incorporation of domain-specific knowledge into image analysis, and taking advantage of the recent development of knowledge representation and inferring techniques, knowledge-based systems to emulate the human way of solving domain-specific problems have been developed

The Naval Research Laboratory began development of the Semi-Automated Mesoscale Analysis System (SAMAS), a comprehensive set of algorithms that handles the entire automated analysis problem, from low-level segmentation through intermediate-level feature formation and into higher-level Artificial Intelligence modules that estimate positions of previously detected features when cloud cover obscures direct observation in the current image set [26]. The SAMAS architecture is shown in Figure 2.

SAMAS, however, only employs feedback from the high-level modules to the medium-level modules. No feedback from the medium to low-level is employed, because our objective is to make the low-level module independent of the domain specific knowledge.

Feature locations produced automatically by SAMAS compare favorably with those from human interpretors. An important part of the SAMAS is the labeling of edges or regions identified by a low-level segmentation routine. By "labeling" we mean assignment of an oceanographic identity to each edge or region fragment. The labeling is based on oceanographic knowledge of some form. The required knowledge could be sea-surface temperature values characteristic of a given feature, spatial context (*e.g.* "north of" or "south of"), a previous analysis, or any other source of knowledge that helps identify features. The main modules of the SAMAS are the segmenter, labeling module, and expert system to predict the position and size of the mesoscale features. In the next section a brief review of the importance of these module in any vision systems is provided.

The current version of SAMAS groups various modules into these three categories. A cloud detection algorithm processes the thermal infrared image of the ocean to classify all pixels either as cloud pixels or non-cloud pixels [Sonia, 1992]. Considering only non-cloud pixels, the system uses the cluster shade texture measure as the low-level operation and detection of zero crossings in cluster shade as the medium-level operation, leading to a set of edge primitives (Holyer and Peckinpaugh, 1989). SAMAS uses two-step nonlinear relaxation [20] to label edge primitives [22]. In the first relaxation labeling step, *a priori* probability values of the edge pixels are computed

using *a priori* knowledge of the approximate sizes and positions of the features, based on a previous analysis (typically from one week earlier). In the second step, these probability values are updates using compatibility coefficients in an iterative fashion, until the values stabilize. The relaxation labeling technique reduces uncertainty in the assignment of labels to edge pixels [22]. A rule based expert system predicts the future position of the mesoscale features. There is a feedback from the expert system to the medium level module. We present some previous methods that have been developed: Neural Networks, Relaxation technique and Expert Systems.

4.2 Neural Networks

Neural networks are composed of many simple computational units that work in unison to process information. A network is not programmed in a traditional sense, rather the weights that connect the neurons are "learned" by the network through repeated applications of input data for which the desired result is known. Neural networks use their massively parallel architecture to explore many competing hypothesis simultaneously. This gives neural networks great potential for application in areas such as speech and image recognition.

The potential benefits of neural networks extend beyond massive parallelism. Neural networks can process inexact, ambiguous, fuzzy data that do not exactly match any information stored in memory. Also, compared to traditional statistical techniques, neural network classifiers are non-parametric and make weaker assumptions about the shapes of the underlying distributions than traditional statistical classifiers.

Neural network technique was applied to the mesoscale labeling problem. The neural network used was a feed-forward, fully connected net with an input layer, one hidden layer and output layer. A supervised training algorithm, back propagation, was used in this neural network. Backpropagation uses a gradient descent search technique to minimize the mean-squared-error between the desired output and the actual network output. In back propagation, input patterns along with the desired output patters are presented to the neural net. In the individual nodes of the hidden and output layers, weights are applied to each input going into the node. These values are then summed and evaluated using a transfer function, sigmoid function. The value fo the transfer function on each node is passed on to the next layer. The process is repeated until the output layer is reached. The value of the output layers is compared with the desired output and the error is computed. This computed error is then propagated back through the network. This entire phase is known as training phase. The goal of the training phase is to minimize the global error of the network by modifying the weights. The phase continues until some user-specified values or constraints is met. once the net has been trained, an independent test using set of inputs is presented to the net, and the output is generated.

4.3 Relaxation Technique

An important research area in image analysis and image interpretation technology is the development of methods that blend contextual information with conventional

image processing algorithms. A literature survey clearly indicates that such a hybrid approach yields good results. Relaxation labeling is one such process. Relaxation labeling has been applied to a variety of image processing problems e.g., linear feature enhancement [59], edge enhancement, image enhancement, pixel classification. A recent survey article by Kittler and Illingworth [20] on relaxation labeling highlights the importance of this area of research. The survey paper also points out the advantages and possible applications of relaxation methods. More importantly, the relaxation labeling approach was elegantly described by Rosenfeld [44] who investigated the problem of labeling the sides of a triangle and proposed a set of schemes to solve the problem. They also concluded with the result that the nonlinear probabilistic relaxation schemes yield better results than the others. Hancock and Kittler have demonstrated that probabilistic relaxation can be successively applied to edge labeling. They also suggest two features that are key to the success of such a labeling system: firstly, a probabilistic framework to represent a world model to ensure internal consistency, and a dictionary of labeling possibilities for the entire context-conveying neighborhood for each object as the second. They have shown that the resulting algorithm combines evidence for the presence of edges by drawing on prior knowledge of the permitted label structures.

The goal of the relaxation process is to reduce the uncertainty (and improve the consistency) in the assignment of one of the labels to each object in a set of related objects. In the oceanographic feature classification problem, the classes are the various oceanographic features, namely the north and south walls of the Gulf Stream, cold eddy, warm eddy, shelf front and coastal boundary. Refer to Figure 1 to

identify the positions of these oceanic features in a typical image. The objects are the individual pixels in a set of registered multi-temporal images. The uncertainty could be due to the cloud cover or the overlap of the features, features not belonging to one of the classes, noise in the image, or other factors. In this paper, we are attempting to label mesoscale features, but the ocean exhibits variability on all spatial scales. Thermal structure on scales smaller than mesoscale will interfere with the mesoscale feature labeling process. The underlying mathematical framework necessary for the relaxation labeling method is described in the following paragraphs.

Let $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_m\}$ be the set of possible labels that may be assigned to each pixel $\mathbf{x}$ in the IR image. Also we let $p_\lambda^k(x)$ denote the probability that the pixel at **x(i,j)** belongs to the object λ after k iterations of the relaxation algorithm.

There are two steps in executing the probabilistic relaxation algorithm. In the first step, a priori probabilities are evaluated with the help of ground truth data, a previous but recent mesoscale analysis or both. In the second step, these a priori probabilities are iteratively updated (relaxation) until a consistent labeling is reached. We now discuss these two steps in detail.

Step 1: Estimating the a priori probabilities

Let $p_\lambda^0(x)$ denote the a priori value, that is, the probability that pixel **x(i,j)** belongs to the object λ at the zeroth iteration. The Bayesian probability equation is used to evaluate this value. The equation (1) is used to calculate $p_\lambda^0(x)$.

$$p_0^\lambda = \frac{p(x \mid \lambda) \;\; P(\lambda)}{\sum_\lambda p(p(x \mid \lambda) \;\; P(\lambda)}$$

where $p(x \mid \lambda)$ denotes the conditional density function and $P(\lambda)$ the probability of occurrence of the object λ.

To evaluate the conditional density function $p(x \mid \lambda)$, a set of parameters is measured at the pixel **x(i,j)**. Let X denote the parameter vector. The following parameters are used to form the vector X:

1. vector from origin to pixel **x(i,j)**, both the magnitude and direction.
2. gray scale intensity value at the pixel **x(i,j)**.
3. the edge magnitude.

For each object, the mean vector μ_λ and the covariance matrix Σ_λ are computed. Also it is assumed that the conditional density function follows a normal distribution. Hence the conditional density function $p(x \mid \lambda)$ is evaluated using equation (2).

$$p(x \mid \lambda) = (2\pi \mid \sum_\lambda \mid)^{-\frac{1}{2}} exp\{-\tfrac{1}{2}(X-\mu_\lambda)^t \sum_\lambda^{-1}(X-\mu_\lambda)\}$$

to compute P(λ), relative areas of the objects are considered. The number of pixels in the object λ is n_λ. Then $P\lambda$ can be calculated using equation

$$P(\lambda) = \frac{n_\lambda}{\sum_\lambda n_\lambda}$$

Step 2: Iterative updating algorithm We now discuss the probability updating rule. The new estimate of the probability of λ at **x(i, j)** is given by

$$p_\lambda^{k+1}(x) = \frac{p_\lambda^k(x)(1+q_\lambda^k(x))}{\sum_\lambda p_\lambda^k(x)(1+q_\lambda^k(x))}$$

where q_λ^k (x) is called the update factor.

The updating factor for the estimate p_λ^k(x) at the kth iteration is given be equation

$q_\lambda^k = \frac{1}{m} \sum_\lambda \sum_{\lambda'} r_{\lambda\lambda'}(x,y) p_{\lambda'}^k(y)$

where m is the number of objects. In this equation, $r_{\lambda\lambda'}(x,y)$ denoted compatibility coefficients. These coefficients are computed as in [44].

According to the relaxation scheme, $r_{\lambda\lambda'}(x,y)$ is a measure of the probabilistic compatibility between label λ on point x and label λ' on point I , and has the following characteristics:

1. If λ on x frequently co-occurs with λ' on y, then $r_{\lambda\lambda'}(x,y) > 0$, and if they always co-occur, then $r_{\lambda\lambda'}(x,y) = 1$.

1. If λ on x rarely co-occurs with λ' on y, then $r_{\lambda\lambda'}(x,y) < 0$ and if they never co-occur, then $r_{\lambda\lambda'}(x,y) =$ -1.

1. If λ on x occurs independently of λ' on y, then $r_{\lambda\lambda'}(x,y) = 0$.

Implementation of the above technique is carried out in two stages. First, the a *a priori* probabilities are estimated using a manually prepared mesoscale analysis on an image obtained 3-5 days prior to the test image. Approximate position of the Gulf Stream and the eddies are given to the labeling program to compute the initial probabilities. In the second stage, the probabilities are updated using compatibility coefficients. The iteration terminates after the probabilities stabilizes. For more details on implementation, please refer Krishnakumar [21] paper. The labeled images of april 17th and april 21st are shown in Figure 16 and 17. Our experiments over a set consisting of twelve images obtained over a period of two months shows that the

Figure 16: Relaxation labeled image on Figure 1

Figure 17: Relaxation labeled image on Figure 11

relaxation labeling module is too biased over the previous knowledge. We developed another technique to reduce the dependence on the previous knowledge. In the next section we discuss this technique.

4.4 Expert System

In the earlier developments of knowledge-based vision systems, there was no sharp division between knowledge based (high level, semantic) models and general purpose (low level, non-semantic) models, partially because the knowledge-based methodology was not available to provide the tools to do so. With the growing need for incorporating domain specific knowledge in image analysis, and subsequent development of knowledge representation and inferring techniques, many expert systems were developed to emulate the human way of solving domain specific problems.

The most popular approaches to representing the domain knowledge (both facts and heuristics) needed for an expert system are production rules, semantic networks,predicate logic, frames and blackboards. Symbolic problem solving methods, such as modus ponens, resolution, and in-exact reasoning have been employed for drawing inferences. The role of the inference engine is to sequence and select the facts in the knowledge base, match the extracted image features against facts, and track inferences. The function of the controller of the inference engine is to decide how to start the inference process using the facts in the knowledge base, how to resolve conflicts and decide which rule to fire. There are three well known controlling strategies: top-down, bottom-up and a combination of the first two strategies.

Top-down strategy is a goal oriented processing, where queries/hypotheses are generated. A query may be subdivided into subqueries. For instance, a query like "Is there any residential area present in the image?" may be generated. The subqueries generated may be like "Are there any roads present in the image?" and "Are there any houses in the image?". These subqueries are again partioned into subqueries and so on. For the subquery "Are there any houses present in the image", the system tries to extract isolated rectangular regions invoking low-level image processing routines. After extraction of these regions, a linear formation of these regions on either side of the road is searched. If all these queries are solved, then the presence of a residential area is inferred, stored, and used for further analysis. An example is the query "Are there any water tanks present in the residential area?". Thus, the creation of a high level query results in the invocation of low-level image processing routines such as region growing, shape analysis, thresholding and other routines. Top-down analysis is extremely useful for vision systems in application where a priori information about the expected image objects and their relationship are available. In such circumstances, there is a possibility of generating meaningful queries and subqueries resulting in an efficient system.

In the bottom-up strategy, the system starts segmenting the image to find homogeneous regions. These regions are analyzed for shapes, intensity values, and their relationship with other regions. These facts extracted from the low-level modules are stored. The knowledge base is searched and matched with the facts to label the image regions. The analysis is driven by the data at the lowest level. It is not advisable to constraint a vision system to work purely in a top-down or bottom-up strategy.

Matsuyama [30] developed an integrated top-down and bottom-up strategy based expert system for aerial photographs. Goodenough [10] developed an expert system using Prolog.

The advent of successful Artificial Intelligence techniques to emulate the human brain, various knowledge representation schemes and control strategies, combined with the lack of precise mathematical models to predict the dynamic behavior of mesoscale features, has led to the investigation of the usefulness of a knowledge-based system to predict the mesoscale features either in a sequence of images or in an image, using a prior knowledge of the sizes and positions of the features in an image obtained 3-4 days prior to the sequence of images. An expert system with a rule base concerning the evolution of mesoscale ocean features in the gulf stream region of the North Atlantic was developed at NRL. The essential features of the system are discussed in brief in succeeding paragraphs.

The first and foremost step in the design of the expert system was the compilation of knowledge pertaining to the formation, movement, evolution and decay of eddies. The knowledge was accumulated through interviews with experts and by reviews of the technical literature [53]. This compilation effectively forms the knowledge base of the system. The next step was to choose an efficient representation scheme.

Rules, semantic nets, frames, blackboards are some of the efficient media of representing facts. Initially, NRL developed a rule-based expert system coded in OPS83 with supporting procedures implemented in the C language. Rules are an appropriate

form of representing unordered knowledge, e.g, an expert knows that a warm eddy rotates clockwise and moves west or southwest, and a cold eddy rotates counterclockwise and moves southwest, with differences related to location and proximity to the Gulf Stream. Currently these facts are represented in the form of rules because the knowledge of the mesoscale features was found to be easier to represent this way.

The rule-based system's working memory stores the status of all known mesoscale events, e.g, size, position, shape, etc, provided by the medium-level analysis to initialize the working memory. For purposes of rule-based computation, the ocean is partitioned into nine disjoint regions, each region having its own set of rules for cold and warm eddies. An eddy's behavior as hypothesized by the expert system depends upon which region its center is in at the beginning of a time step. The expert system has different rules for ring and Gulf Stream behavior in each of the nine geographical regions. Basically, the motion has a region-dependent velocity vector. However, a ring that is closer than a certain critical distance from the Gulf Stream undergoes a modification of the basic motion. The Gulf Stream interaction rules are also region-dependent. The details depend upon how close the ring is to the Gulf Stream; the Gulf Stream interaction may result in a deflection or a looping motion, with possible coalescence with the Stream in some cases. Ring sizes decrease with time. The rate of decrease depends on the region and on whether there is Gulf Stream interaction. A ring that shrinks below a certain size disappears. The reader is recommended to refer to Thomason [1989] for details of the rules and the architecture of the system. Using the initial status of the rings, the expert system updates their status.

The knowledge base of the expert system has a very simple structure. The system models the movement of the two types of eddies in each of the nine different regions of the Gulf Stream. The knowledge base consists of two rules for each type of eddy in each region, giving a total of 36 rules. Each rule has a very simple left-hand side (LHS) that identifies the type of eddy and the region. The two rules for the type of eddy in the region where its center is located are always fired. The solution strategy is slightly different for cold and warm eddies.

The first rule for cold eddies in each region estimates the new radius for the eddy and asserts a fact into working memory that causes the second rule to fire for that eddy. The second rule also has a very simple set of patterns on the LHS, but the right-hand side (RHS) is a very long set of decisions and calculations. This procedural code on the RHS determines the distance of the eddy from the Gulf Stream and calculates a ratio that specifies the degree of intersection of the eddy and the Gulf Stream. The change in latitude and longitude of the eddy are then predicted based on the degree of interaction. If the interaction is negligible, then the calculation is very straightforward. If there is significant interaction, then the degree of interaction is used to select an interaction regime that is used for a more complex set of calculations. Depending on the interaction regime selected, changes may be made in the original value calculated for the radius, the direction of movement of the eddy, and the speed at which the eddy travels. The rules for each region are very similar, differing primarily in the parameter values used in the calculations.

The first rule for warm rings in each region not only estimates the new radius but also estimates the new longitude and latitude for the eddy. The RHS of the first

rule may also include constraints that model the limitation of the movement of the eddies by physical barriers that occur in specific regions. The second rule estimates the degree of interaction with the Gulf Stream based on the revised radius, longitude, and latitude and then makes revisions in the estimated radius and position based on the degree of interaction. Again, most of the decisions are made on the RHS of the second rule in complex procedural code that is repeated in the rules for each region.

Chapter 5

Topography-Based Feature Labeling

Detection of topographic structures in a digital image is not a new idea. There has been a wide variety of techniques described to detect pits, peaks, ridges and ravines. Peuker and Johnston [39] characterize the surface shape by the sequence of positive and negative differences as successive surrounding points are compared to the central point. Peuker and Douglas [40] describe several variations of this method for detecting one of the shapes from the set (pit, peak, pass, ridge ravine, break, slope , flat). They start with the most frequent feature (slope) and proceed to the less frequent, thus making it an order-dependent algorithm.

Johnston and Rosenfeld [19] attempt to find peaks by finding all points P such that no points in a n-by-n neighborhood surrounding P have greater elevation than P. Pits are found in an analogous manner. To find ridges, they identify points that are

either east-west or north-south elevation maxima. This is done using a "smoothed" array in which each point is given the highest elevation in a 2 x 2 square containing it. East-west and north-south maxima are also found on this array. Ravines are found in a similar manner.

Paton [36] uses a six-term quadratic expression in legendre polynomials fitted to a small disk around each pixel. The most significant coefficients of the second-order polynomial yield a descriptive label such as constant, ridge, valley, peak, saddle.

Grender's [11] technique compares gray level elevation of a central point with surrounding elevations at a given distance around the perimeter of a circular window; the radius of the window may be increased in successive passes through the image.

Toriwaki and Fukumara [55] take a totally different approach from all the others. They use two local features of gray level pictures, connectivity number, and coefficient of curvature for classification of the pixel into peak, pit, ridge, ravine, hillside.

The above mentioned techniques can be classified into two categories. In the first category, classification is performed by operations directly on the discrete image, whereas in the latter, a continuous surface of a certain kind is locally fitted at each pixel first, and classification is achieved based on this approximation of the underlying surface.

We find many techniques in the literature to extract topographic labels. However, techniques need to be developed for grouping and assembling topographically labeled pixels to build primitive features needed in solving practical computer vision problems. In an independent study, notable researcher Theo Pavlidis also stressed the

need for grouping the topographic labels to build character recognition algorithms. He designed several algorithms using the topographic labels and showed improved performance in extracting characters from documents.

The digital image is an equal interval grid sampling of real values of a continuous function, f In each neighborhood of the image the underlying gray tone intensity function, f can be parameterized as a polynomial in row, r and column c coordinates. A brief summary of the mathematics of polynomial fitting within a neighborhood follows. The reader is referred to Haralick [14] for a complete discussion. Each neighborhood can be expressed as a cubic polynomial in r, c space.

$$f(r,c) = k_1 + k_2 r + k_3 c + k_4 r_2 + k_5 rc + k_6 c_2 + k_7 r_3 + k_8 r_2 c + k_9 r c_2 + k_{10} c_3.$$

Haralick has shown that the coefficients $k_1, k_2, ., ., ., k_{10}$ can be expressed as linear combinations of the intensity values within the image neighborhood. Therefore, the coefficients can be calculated by convolution of the image with a kernal of appropriate weights. The required convolution kernals for fitting a polynomial to a 3 x 3 pixel neighborhood are shown in Figure 18.

After fitting, the local intensity function, f, has been fit with a polynomial, first and second derivatives of the surface can be calculated. If we evaluate the derivatives at the center of the neighborhood, *i.e.*,(r,c) coordinates = (0,0), the following relationship between derivatives and fitting coefficients applies.

$$\delta f / \delta r = k_2$$

$$\delta f / \delta c = k_3$$

$$\delta^2 f / \delta r^2 = 2k_4$$

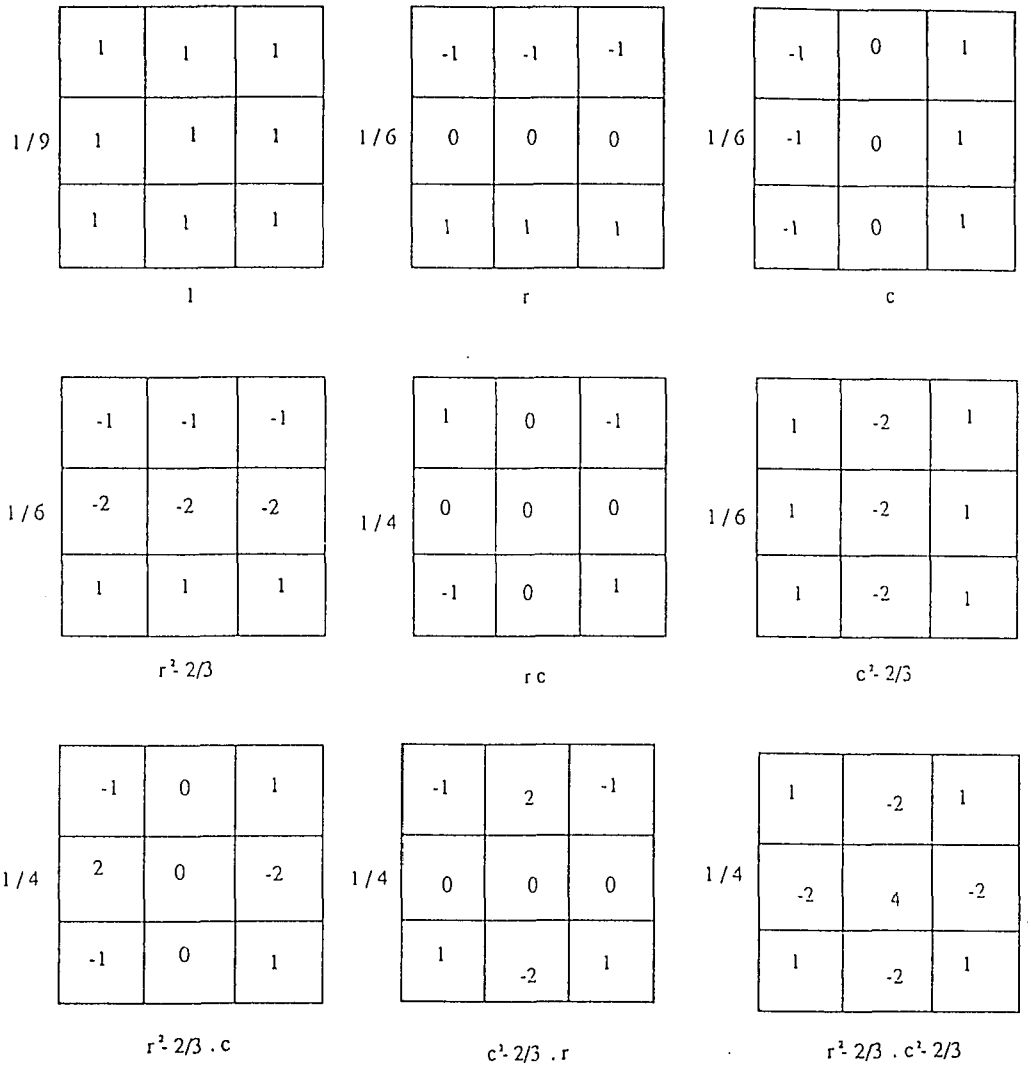


Figure 18: Nine 3 x 3 masks for computing the coefficients

$\delta^2 f/\delta c^2 = 2k_6$

$\delta^2 f/\delta r \ \ \delta c = k_5$

The magnitude of the gradient vector is then $\sqrt{{k_2}^2 + {k_3}^2}$ and its direction is $tan^{-1}(\frac{k_2}{k_3}$. The second directional derivatives may be computed using the *Hessian* matrix,H, defined as

$$\begin{pmatrix} \delta^2 f/\delta r^2 & \delta^2 f/\delta r \ \ \delta c \\ \delta^2 f/\delta c \delta r & \delta^2 f/\delta c^2 \end{pmatrix}$$

We know

$$\delta^2 f/\delta c \ \delta r = \delta^2 f/\delta r \ \delta c$$

The two eigenvalues of H, λ_1 and λ_2, are the values of the extrema of the second directional derivatives, and their associated eigen vectors are the directions in which the second directional derivatives are extremized (Haralick 1984).

Based on the directional derivative information, the surface, f, in a neighborhood can be classified into one of several topographic shapes such as ridge, valley, flat, saddle, convex hill, concave hill, etc (Haralick *et.al*, 1983). In the present application, we are interested only in flat, concave hill and convex hill topographic classes.

5.1 Topographic Classification Scheme

To classify the topographic shape at each pixel as either flat, convex hill, or concave hill, the scheme in Table 4 is used. In the table the symbol "+" means significantly greater than, "-" means significantly less than, "+=" means significantly greater

than or equal to and "-=" means significantly less than or equal to. Thus, "+=0" is read as significantly greater than or equal to zero. The symbol "*" denotes all other possible values. In this case no class is assigned. For example, let us consider a pixel neighborhood and let the computed values of gradient magnitude and second directional derivatives λ_1 and λ_2 at center of the neighborhood (0,0) be 10, 7 and 3 respectively. Then the pixel may be classified as Convex hill. If the computed values 12, -7 and -1, then the pixel is topographically labeled as Concave hill. If the values are 0, 0 and 0, then we can label it as a flat. Note that a cloud mask is available to the topographic classification routine which also assigns no class if any pixel within the neighborhood is flagged as cloud covered. For example, if the center of the pixel neighborhood is a cloud pixel, then "No label" is assigned.

5.2 Labeling the Gulf Stream

When the (Warm) eddy is near the (north wall of the) gulf stream, the situation poses considerable problems to the relaxation labeling module. The relaxation labeling module ambiguously labels some eddy pixels as Gulf Stream pixels and some Gulf Stream pixels as warm eddy pixels. The primary reason is that relaxation technique initializes the probability values of the edge pixels based on the distance of the edge pixels from the previous position of the features. This **a priori** probabilities values plays a vital role in the final results of the labeling module. The relaxation module emphasizes more importance on the previous position of the features. In doing so, the final results seems to be more baised towards the previous positions of the features. In

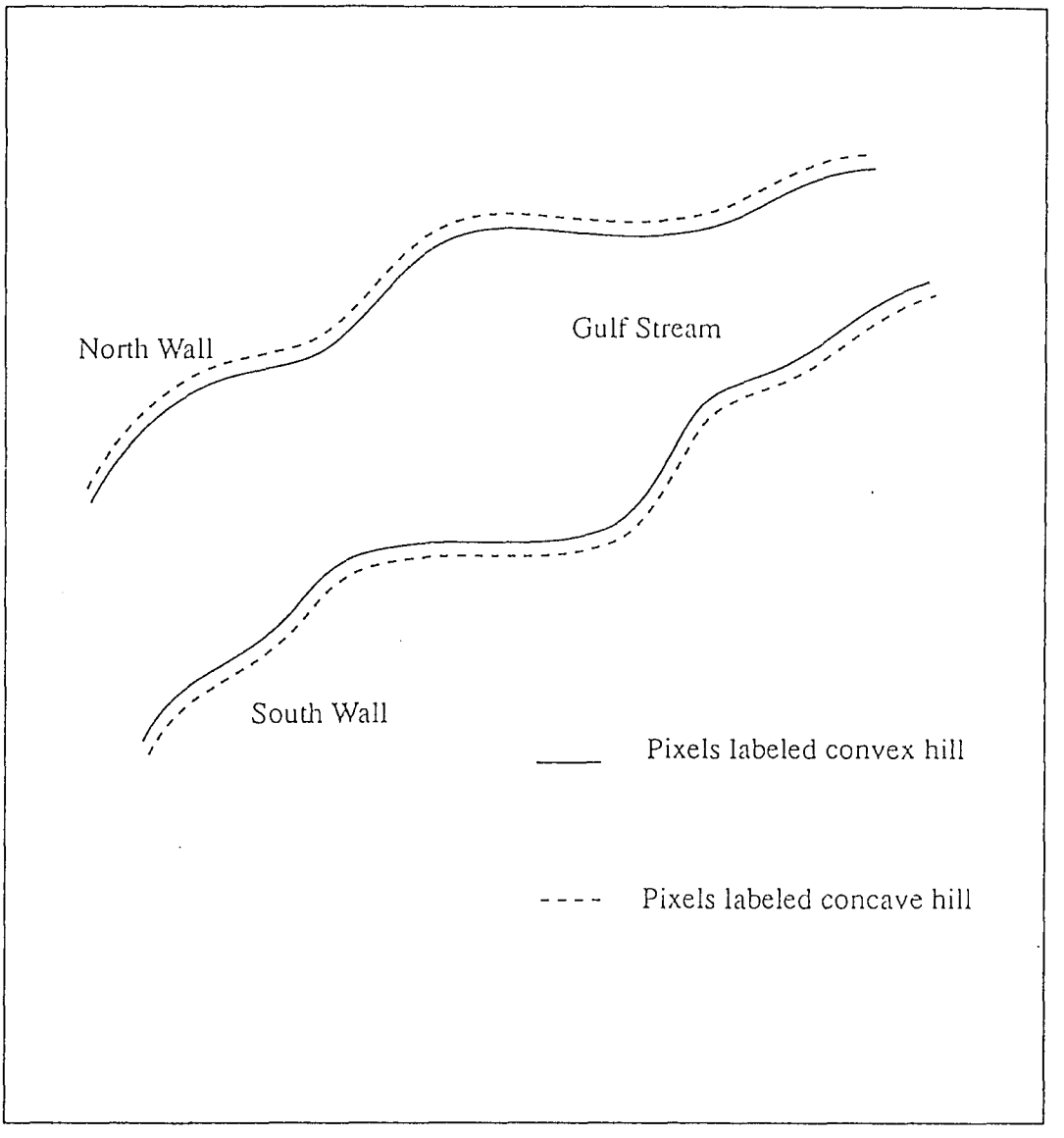


Figure 19: Position of the convex and concave hill pixels near the Gulf Stream

other words, the error in the results of the relaxation module is directly proportional to the errors in the previous analysis. We anticipate that such errors would be reduced in our topography approach.

Experiments with a number of images have shown that pairs of pixels, one classified as convex hill and the other as concave hill occur along the north and south walls of the Gulf Stream as shown in Figure 19. Spurious concave/convex pairs occur elsewhere in the image also, but these can be eliminated by a local correlation criterion. We can also remove spurious pixels by considering only those chain of pixels which are longer than 10 pixels. All pixels in a chain are either convex hill pixels or concave hill pixels. A chain of pixels may be either a line, curve, arc or any other geometric entity. Detection of these entities is tedious and cumbersome. Thus we employed a connected component labeling and counting algorithm to extract the chain of pixels. Thus, pairs of convex hill and concave hill classified pixels are examined initially as possible North wall points. The criteria of examination is the gray level intensity values. Possible North Wall points are then compared with the position of the Gulf Stream in the previous analysis and eliminated if the distance exceeds some specified value. It is also noted the gradient magnitude and the second directional values, λ_1 and λ_2, should have very high values at the North Wall. Table 5 presents the criterion for using these values to assign North Wall labels to a pixel. In the Table 5 the symbol "++" or "−" mean very large positive and very large negative values, respectively. For example, if the computed values of gradient magnitude and second directional derivatives λ_1 and λ_2 at center of the neighborhood (0,0) are 20, 10 and 4, then the convex hill pixel is examined for a possible North Wall pixel. Similarly, if the values

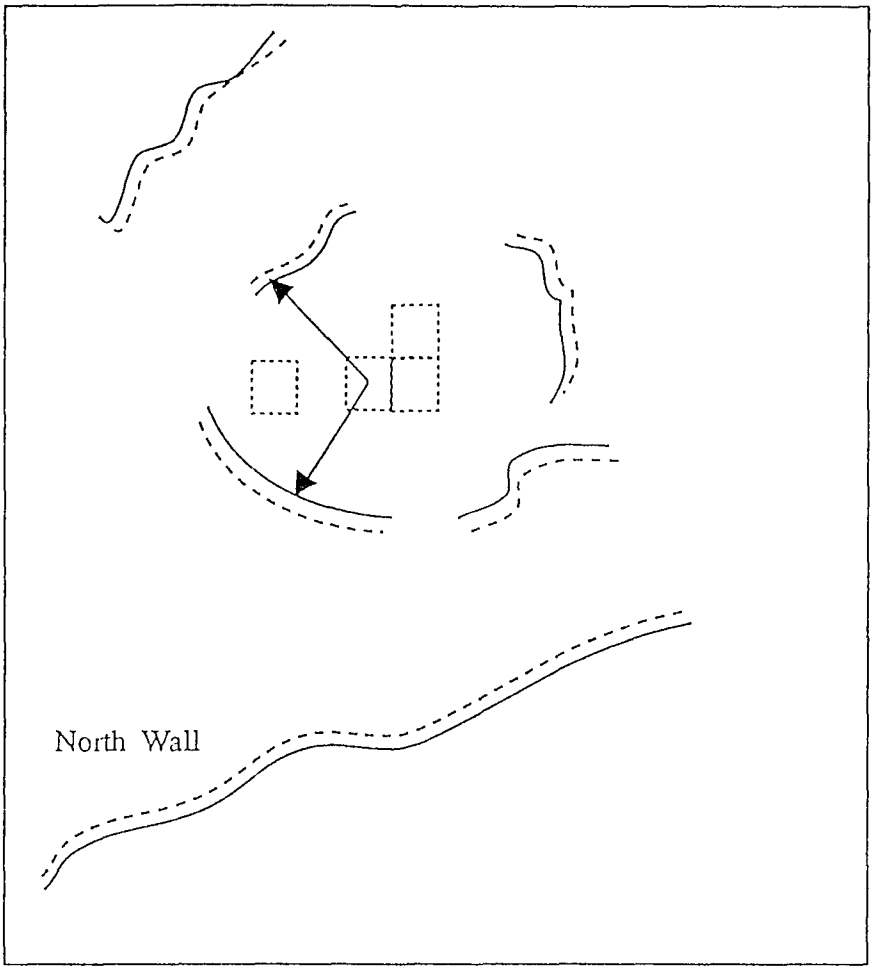
are 18, -13 and -3, again the Concave hill pixel is examined for a possible presence of North Wall. Labeling of the South Wall is similar to the North Wall except that the entries in the classification table (Table 5) are changed slightly to indicate the general decrease in the intensity gradient at the South Wall compared to the North Wall.

When the features are (fully or) partially hidden under the clouds, the labeling technique suffers yet another situation. The effects of feature proximity and cloud cover are considerably reduced using the topography based approach. This is primarily due to the initial classification of the raw data into topographic labels.

5.3 Labeling of Warm Eddies

A warm eddy is an area of relatively uniform low intensity surrounded by higher intensities in adjacent regions. Most eddies are not simple shapes. In practice the rotational dynamics of the eddy result in shear across the periphery and the resulting intensity spatial distribution patterns can become very complex. However, most eddies will contain significant numbers of interior pixels that classify as topographically flat. Pixels at eddy boundaries normally classify as convex or concave hill.

If one started at the flat interior pixels of an eddy and proceeded radially, normally when the eddy boundary is encountered, a convex hill pixel would be encountered first, followed immediately by a concave hill. This characteristic property of radial rays is utilized to distinguish eddies from the Gulf Stream. The ray firing procedure begins with an m x m box centered on the position of the eddy in the previous analysis.



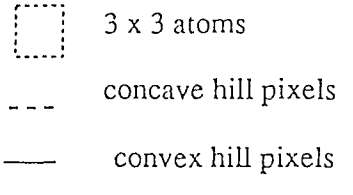


Figure 20: Probes originating from the center of 3 x 3 flat atoms

Within the m x m window all 3 x 3 areas of flat topographical class are considered as possible eddy centers (called as atoms here). Each atom is given a counter which is initialized to zero. From these atoms rays are sent radially in 36 directions. The procedure is illustrated in Figure 20. Rays follow line path based on Bresenham's line drawing algorithm [2]. Whenever a ray first intersects a pixel classified as convex hill and then immediately strikes a concave hill pixel, the counter for that atom is incremented. The ray is then moved to the next direction and the procedure is repeated. If the probe does not intersect a convex hill pixel within a specified distance (related to typical eddy size) the path of that ray is terminated without incrementing the atom counter. After sending rays from all atoms and saving an accumulator for each, the accumulators are examined for the high values indicating that the atom is a possible eddy center. Any edges (*i.e.* convex/concave pairs) "visible" from the candidate eddy centers are labeled as eddy pixels. If two candidate eddy centers are close together, one could be selected and the other discarded based on some criterion such as proximity to the eddy position in the previous analysis, or the atoms could be merged into one atom located at the center of gravity of the initial atoms.

5.4 Implementation Results

The edge detection and labeling capabilities of the topographical approach developed here have been evaluated on several test images of the Gulf Stream. Polynomial fitting window sizes 3 X 3, 5 X 5, 7 X 7 were evaluated. For a window size 7 X 7, the topographic classification scheme produced good results. In all of the labeled images,

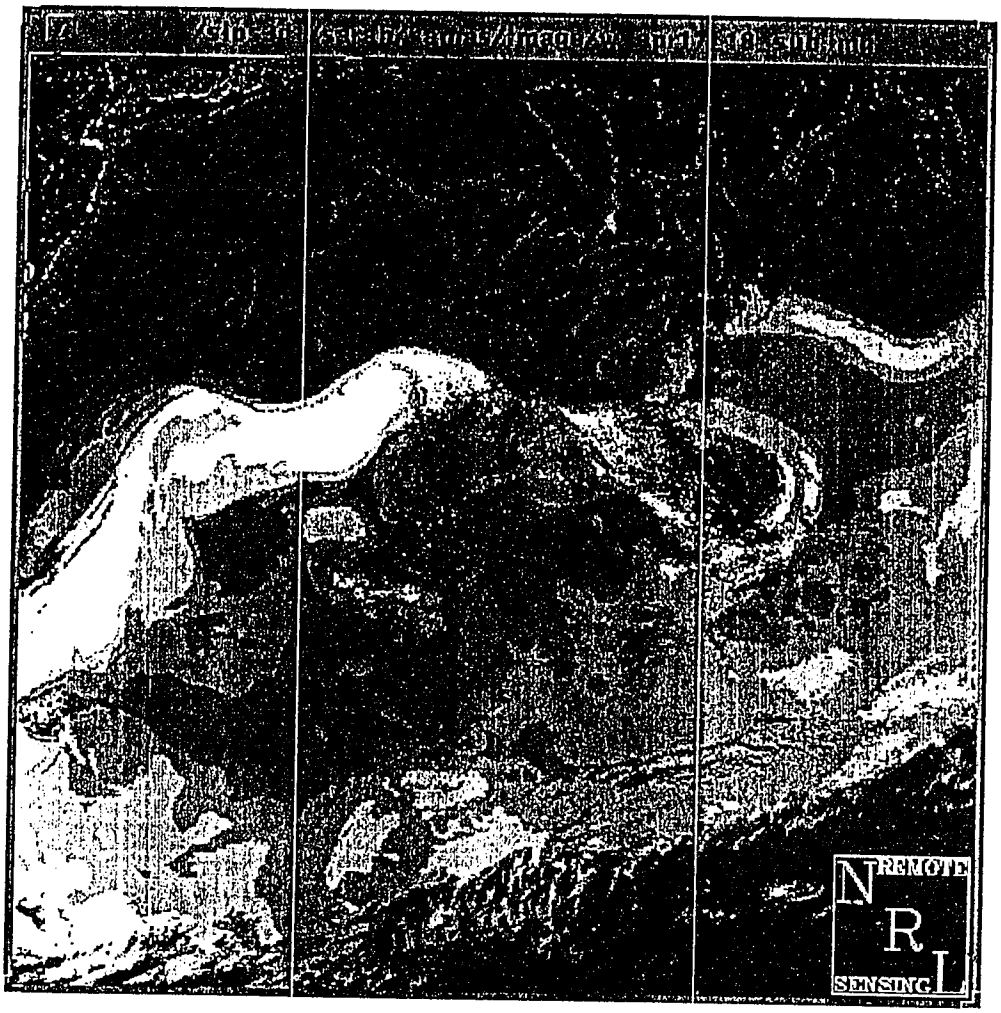


Figure 21: Convex hill pixels superimposed on Figure 1

Figure 22: Concave hill pixels superimposed on Figure 1

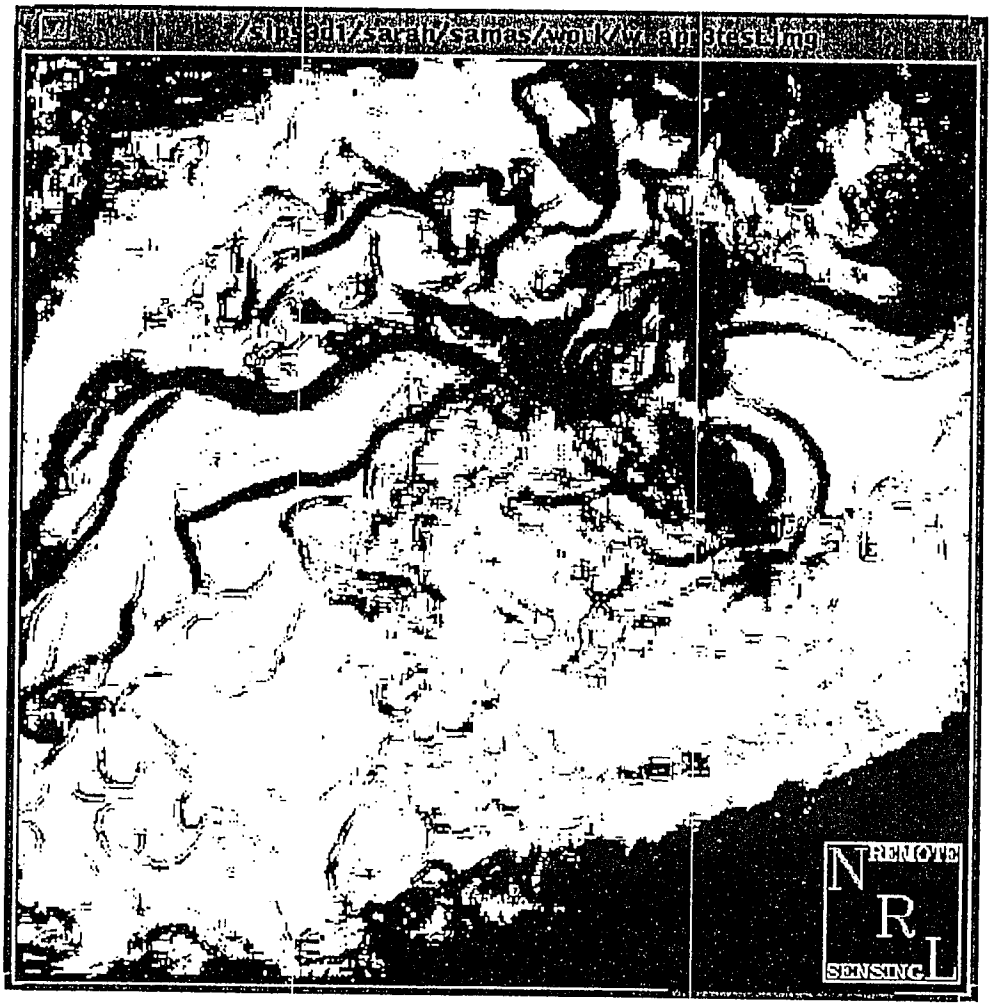


Figure 23: Flat pixels superimposed Figure 1

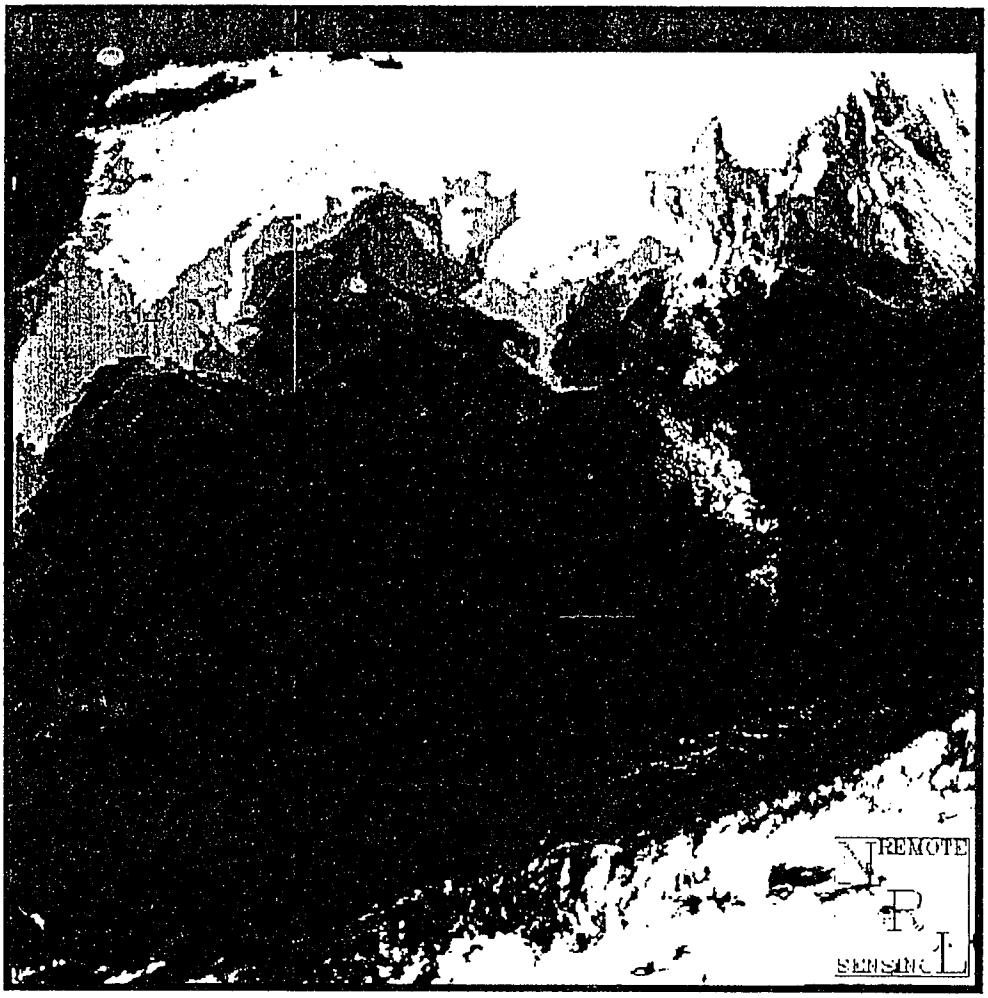


Figure 24: Pixels labeled as North Wall and Warm Eddy in April 17 image

Figure 25: Pixels labeled as North Wall and Warm Eddy in Figure 11

Figure 26: Pixels labeled as North Wall and Warm Eddy in Figure 12

we used a window size of 7 x 7. The technique is being tested on a number of test images. Only chains of length 10 or more Convex hill and Concave hill pixels are considered in order to eliminate spurious pixels. In Figure 21, the convex hill pixels are superimposed on the original image shown in Figure 1, the concave hill pixels are shown in Figure 22, and the flat pixels are shown in Figure 23. Figure 24 show the pixels labeled as North Wall and as Warm Eddy. The labeled North Wall and Warm Eddy of the Gulf Stream for Figures 11 and 12 are shown in Figures 25 and 26. The North Wall pixels are extracted from the chain of convex and concave hill pixels by considering only intensity values of the pixels. However, by constructing a histogram of a sub-window centered at a pair of concave hill and convex hill pixel, we can examine the presence of a North Wall using a local thresholding technique. We encountered some difficulties in segmenting the South Wall and the Cold Eddy of the Gulf Stream because of very low gradient values associated with these features. At present, we are trying to the combine the flat regions and the North Wall of the Gulf Stream to extract the South Wall.

Chapter 6

Conclusions and Future Directions

In the first part of the dissertation, we presented a Histogram-Based morphological edge detector (HMED) blending the cues of image histogram with the morphological operations. The new edge detector provided improved performance while being conceptually simple and computationally efficient. The interesting feature of HMED is that it bridges two edge detection methods: histogram based methods search for the *valleys and peaks* in the histogram, and the morphological methods (BMM and ATM) search for the *extreme* intensities that have non-zero histogram heights. The motivation of designing HMED is due to the fact that the previous morphological edge detectors are designed to work only in the image domain. Such designs ignore the vital information contained in the histogram of an (sub-)image. As a consequence, various weak gradient values pertaining to important features are missed in oceanographic IR images. HMED is designed to operate on the histogram of the image to extract the weak gradients. Thus, we defined new morphological operations defined over the histogram of a neighborhood of a pixel.

We implemented HMED on several high-performance interconnection networks: tightly coupled MASPAR (SIMD machine) and loosely coupled distributed network of workstations (client-server model). We also presented implementation results of HMED on both models of architectures.

In the second part of the disseration, we proposed a general computational framework to address the problem of labeling oceanographic images. The essential ideas stem from fitting a bicubic polynomial to each pixel's neighborhood and assigning topological labels based on the first and second directional derivatives of the polynomial surface. The relationship between the oceanographic features in infrared satellite imagery and the topographic structures is also presented. Our technique detects the North Wall of the Gulf Stream and its associated Warm Eddies. We also introduced a new algorithm to combine edge and region information for the detection of eddies. The Warm eddies are easily extracted by this new technique. The most significant improvement obtained by first topographically classifying the image and forming correlation between topographic labels and oceanographic features is the decreased dependency on the previous analysis of the features (typically one week earlier). An added advantage of the classification scheme is that it can also be applied to detect edges.

We have presented various techniques that exist in the literature to solve the problem of labeling of mesoscale features. We emphasize more research work is needed to improve the automatic interpretation schemes available today. This may be attempted by injecting more knowledge about the scene under investigation, developing

powerful expert systems to assist the interpreter, and by developing new techniques for automatic image interpretation.

The problem of automatic interpretation of natural scenes is quite complex and more powerful techniques are necessary to solve this problem. Application of artificial intelligence is shown to be one of the ways to solve this difficult problem. We emphasize the need for fusing both low-level and high-level knowledge usually structured as a hierarchy of models. Expert systems certainly constitute promising tools for image processing and interpretation.

In the two methods described earlier, namely Relaxation labeling technique and the Topography based approach, the main characteristic nature of the eddies is ignored. For instance, the eddies are predominantly either circular or elliptic or some irregular shape. The extracted edges have to be grouped into smaller straight line segments. Then the line segments should be grouped and tested for the presence of eddies. Thus these techniques should incorporate the knowledge of the shape in order to achieve better and reliable results.

Motion analysis from a sequence of time-varying images is a principal requirement for an automated machine to interpret the image. Motion analysis techniques track an object in a sequence of images using raw gradient-based schemes, edge gradient schemes, or feature tracking schemes. Raw gradient schemes compute visual motion directly from the ratio of temporal to spatial image irradiance gradients. Edge gradient schemes compute visual motion by matching zero crossings found in the sequence of image. Feature tracking schemes compute motion by extracting features in one image and search these features in the next image. Most of the motion analysis

techniques assume rigid motion of the features. However, in oceanographic images, mesoscale features are time varying, *i.e.*, mesoscale features invariably change their position, size and shape. The problem, sometimes, is compounded by the presence of cloud cover. Thus, the first two schemes can be rejected directly because the topology of the edges is not preserved in consecutive images.

The applicability of the feature tracking scheme on these images with and without cloud cover has to be studied. Wu [57] computed the advective velocities and their direction from AVHRR data using temperature feature-tracking approach based on pattern-matching method. Their method is based on finding maximum cross-correlation values within two templates of size 32x32 from two consecutive images. The maximum cross-correlation approach is not suited for the application such as the one considered in this paper because the mesoscale feature changes their direction rapidly, in addition to the presence of cloud cover.

TABLE 4. Topographic Classification Scheme

Gradient magnitude	λ_1	λ_2	Label
+0	+0	+=0	convex hill
+0	-0	-=0	concave hill
0	0	0	flat
*	*	*	no label

TABLE 5. Topographic Labeling Scheme

Gradient magnitude	λ_1	λ_2	Label	Mesoscale features
++0	++0	+=0	convex hill	North Wall
+0	-0	-=0	concave hill	North Wall

Bibliography

[1] D. Argialas. and C. Harlow, "Computational Image Interpretation Models: An Overview and Perspective", *Photogrammetric Engineering and Remote Sensing,* 56(6), 871-886, June 1990.

[2] Bresenham, J.E., "Algorithm for Computer Control of a Digital Plotter", *IBM Systems Journal*, 4(1), 25-30, 1965. .

[3] J.-F. Cayula, P. Cornillon, R. Holyer and S. Peckinpaugh, "Comparative Study of two recent edge-detection algorithms designed to process sea-surface temperature fields", *IEEE Transactions on Geosci. Remote Sensing,* 29, No 1, 175-177, Jan 1991.

[4] J.-F. Cayula and P. Cornillon, "Edge Detection applied to Sea Surface temperature Fields", *Proc. SPIE Tech. Symp. Optical Engg. and Photonics in Aerospace Sensing,* 1990.

[5] Chan, R. and Siu, W.C, "New parallel Hough transform for circles", *Iee proceedings. e, computers and digital techniques* SEP 01 v 138 n 5 Page: 335, 1991.

[6] R.E. Coulter, "Application of the Bayes Decision Rule for Automatic Water Mass Classification from Satellite Infrared Data", *Proc. 17th Int. Symposium on Remote Sensing of Environment,* Vol II, 1983, 589-597.

[7] F. C. Fuglister, "Cyclonic rings formed by the Gulf Stream 1965-1966", *Studies in Physical Oceanography: A tribute to George Wust on his 80th Birthday,* A. Gordon, ed., Gordon and Breach, 1972, pp 137-168.

[8] D.J. Gerson and P.Gaborski, "Pattern Analysis for Automatic Location of Oceanic Fronts in Digital Satellite Imagery", *Naval Oceanographic Office,* TN 3700-65-77, October, 1977.

[9] D.J. Gerson, E. Khedouri and P. Gaborski, "Detecting the Gulf Stream from Digital Infrared Data Pattern Recognition", *The Belle W. Baruch library in Marine Science,* No.12, 1982, 19-39.

[10] D. Goodenough, Goldberg.M., Plunkett.G., and Zelek.J., "An Expert System for Remote Sensing", *IEEE Transaction on Geoscience and Remote Sensing*, 25(3), 1987.

[11] Grender, G. C., " TOPOIII: A fortran program for terrain analysis", *Comput. Geosci*, 2:195-209, 1976.

[12] A. R. Hanon and E. M. Riseman, "Segmentation of Natural Scenes", *Computer Vision Systems,* Academic Press, pp 129-163, 1978.

[13] R. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives", *IEEE PAMI*, vol-6, No 1, 1984.

[14] R. Haralick, L. Watson and T. Laffey, "The Topographic Primal Sketch", *The Intl. Journal of Robotics Research*, Vol-2, No 1, 1983.

[15] R. Haralick, S. Sternberg and X. Zhuang, "Image Analysis Using Mathematical Morphology", *IEEE PAMI,* v-9, 4, July 1987.

[16] R.J. Holyer and S.H. Peckinpaugh, "Edge Detection Applied to Satellite Imagery of the Oceans", *IEEE Trans. GeoScience and Remote Sensing,* vol 27, No 1, pp 45-56, Jan 1989.

[17] James Lee, R. Haralick, and L. Shapiro, "Morphologic Edge Detection", *IEEE Journal of Robotics and Automation,* V-3, No 2, April 1987.

[18] M.F. Janowitz, "Automatic Detection of Gulf Stream Rings", *Technical Report TR-J8501, Office of Naval Research, June 1985.*

[19] Johnston, E.G and Rosenfeld. A, "Digital detection of pits, ridges and ravines", *IEEE Transaction Systems, Man and Cybernetics*, pp:472-480, July 1975.

[20] J. Kittler and J. Illingworth, "Relaxation Labeling Algorithms - A Review", *Image and Vision Computing,* 206-216, 1985.

[21] N. Krishnakumar, S.Sitharama Iyengar, Ron Holyer and Matthew Lybanon, "Feature Labeling in Infrared Oceanographic Images", *Image and Vision Computing,* 8(2):141-147, 1990.

[22] N. Krishnakumar, S.Sitharama Iyengar, Ron Holyer and Matthew Lybanon, "A Technique for feature labeling in infrared oceanographic images", *Fifth Int. Conf. on IIPS,* 368-375, Jan 1989.

[23] Krishnamurthy. S., S. S. Iyengar, R. Holyer and M. Lybanon, "Histogram-Based Morphological Edge Detector," **IEEE Transactions on Geoscience and Remote Sensing**, 32, 4, pp:759-767, July, 1994.

[24] Krishnamurthy. S., S. S. Iyengar, R. Holyer and M. Lybanon, "Topography-based Feature labelling for IR oceanographic images," **Pattern Recognition Letters**, 14, pp. 915-925, Nov 1993.

[25] Krishnamurthy. S., S. S. Iyengar, R. Holyer and M. Lybanon, "Morphological Edge Detection in Infrared Oceanographic Images," *in the Proceedings of the SPIE : 22nd AIPR Workshop* , Vol 2103, pp 2-13, 1993.

[26] M. Lybanon, J.D. McKendrick, R.E. Blake, J.R.B. Cockett and M.G. Thomason, "A Prototype Knowledge-based System to Aid the Oceanographic Image Analyst", *SPIE Vol 635- Applications of Artificial Intelligence III,* 1987, 203-206.

[27] M. Lybanon and J.D. Thompson, "The Role of Expert Systems in Ocean Engineering", *Conf. Proc, An Ocean Cooperative: Industry Government Academia,* 1990.

[28] D. Mason, I. J. Lauder, D. Rutoritz and G. Spowart, "Measurement of C-bands in human chromosomes, *Computer. Biol. Med*, 179-201, 1975.

[29] G. Matheron, "Random Sets and Integral Geometry", *Wiley and Sons,* New York, 1975.

[30] T. Matsuyama, "Knowledge-Based Aerial Image Understanding Systems and Expert Systems for Image Processing", *IEEE Transactions on Geoscience and Remote Sensing,* 25(3), 1987.

[31] D. Mckeown, W. Harvey, and J. Mcdermott., "Rule-Based Interpretation of Aerial Imagery", *IEEE Trans. PAMI,* 7(5), sep 1985.

[32] H. Muammar and M. Nixon, "Approaches to extending the Hough transform", *Proc, ICASSP* , 1556-1559, 1989.

[33] M. Nagao, "Control Structures in pattern analysis", *Pattern Recognition, vol-17,* 45-56, 1984.

[34] D. G. Nichol, "Autonomous Extraction of Eddy-like Structure from Infrared Images of the Ocean", *IEEE Trans. GeoScience and Remote Sensing,* 25, 28-34, 1987.

[35] D. B. Olson, R. W. Schmitt, M. Kennelly, and T. M. Joyce, "A two-layer diagnostic model of the long-term physical evolution of warm-core ring 82B", *Journal of Geophysical Research*, vol-90, no C5, pp 8813-8822, 1985.

[36] Paton. K., "Picture description using Legendre polynomials", *computer Graphics Image Processing*, 4(1), 40-54, 1975.

[37] S. Pelag, J. Naor, R. Hartley and D. Avnir, "Multiple Resolution Texture Analysis and Classification", *IEEE Trans. PAMI, 1984* 518-523, 1984.

[38] S. Pelag and A. Rosenfeld, "A Min-Max Medial axis Transformation", *IEEE Trans. PAMI*, 206-210, 1981.

[39] Peuker, T. K., and Johnston, E. G, " Detection of Surface-specific points by local parallel processing of discrete terrain elevation data", Tech. Report. 206, College park, Md.: University of maryland Computer Science center, 1972.

[40] Peuker, T. K., and Douglas, D. H., "Detection of Surface-specific points by local parallel processing of discrete terrain elevation data", *computer graphics and image processing*, 4(4):375-387, 1975.

[41] W.k. Pratt, *Digital Image processing*, New York: Wiley, p.489, 1978.

[42] Richard Feehs and Gonzalo Arce, "Multidimensional Edge Detection", *Proceedings of SPIE, Visual Communications and Image Processing II,* vol 845, 285-292, 1987.

[43] P. L. Richardson, "Gulf Stream Rings", in Eddies in Marine Science", *A. R. Robinson, Springer-verlag,* pp 65-68, 1983.

[44] A. Rosenfeld, R.A. Hummel and S.W. Zucker, "Scene Labeling by Relaxation Operations", *IEEE Trans. Syst. Man Cyber*, 6, 420-434, 1976.

[45] A. Rosenfeld and A. Kak, *Digital Picture Processing*, " Academic Press", 1982.

[46] The Ring Group, "Gulf Stream Cold-Core rings: their physics, chemistry, and Biology", *Science,* vol 212 pp 1091-1100, 1981.

[47] J. Serra, "Image Analysis and Mathematical Morphological", *Acad. Press,* New York, 1982.

[48] Stanley R. Sternberg, "Grayscale Morphology" *CVGIP*, 35, 333-355, 1986.

[49] P.K. Sahoo, S. Soltani, and A. K. C. Wong, "A Survey of Thresholding Techniques", *CVGIP*, 233-260, 1988.

[50] H. Stommel, "The Gulf Stream: A Physical and Dynamical Description", *2nd edition, Berkeley: University of California Press,* 1965.

[51] Sonia Gallegos, J. Hawkins, and C. F. Cheng, "A New Automated Method of Cloud Masking For AVHRR Full Resolution Data Over The Ocean", *IEEE Geoscience and Remote Sensing,* April, 1993.

[52] J.M. Tenenbaum and H.R. Barrow, "Experiments in interpretation-guided segmentation", *Artificial Intelligence,* 8, pp 241-274, 1977.

[53] M. G. Thomason, and R. Gonzalez, "Database Representation in Hierarchical Scene Analysis", *Progress in Pattern Recognition,* L. Kanal and A. Rosenfeld, editors, North-Holland, Amsterdam, vol 1, 1981.

[54] M. G. Thomason, "Knowledge-Based Analysis of Satellite Oceanographic Images", *International Journal of Intelligent Systems,* V-4, 143-154, 1989.

[55] Toriwaki, J., and Fukumura, T. "Extraction of structural information from grey pictures", *comput. graphics Image Processing*, 7(1):30-51, 1978.

[56] M. Werman, and S. Pelag, "Min-Max operators in Texture Analysis", *IEEE PAMI*, 730-733,Nov 1985.

[57] Q. X. Wu, D. Pairman, S. J. McNeill, "Computing Advective Velocities from Satellite Images of Sea Surface Temperature", *IEEE Transactions on Geoscience and Remote Sensing*, Vol 30, No 1, Jan 1992.

[58] S.W. Zucker, R.A. Hummel and A. Rosenfeld , "An Application of Relaxation Labeling to Line and Curve Enhancement", *IEEE Trans.Computer,* 26, 922-929, 1977.

[59] S. Zucker, "Relaxation Labeling and the Reduction of Local Ambiguities", *Proceedings of the Third International Joint Conference on Pattern Recognition,* 1976.

Vita

Sankar Krishnamurthy received his B.Tech degree at the Regional Institute of Technology, India, in 1986, and holds two MS degrees from Louisiana State University, Baton Rouge, LA., one in Engineering Science and the other in Computer Science. His research interests are in Computer Vision, Computer Graphics, Visualization and Multimedia.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Sankar Krishnamurthy

Major Field: Computer Science

Title of Dissertation:

Tracking Dynamic Features in Image Sequences

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

19th September 1994